

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
No department name is given

Aplikace rozšířené reality pro OS Android

Application of Augmented Reality for OS Android

Zadání diplomové práce

Student:

Bc. Jan Hruška

Studijní program:

N2647 Informační a komunikační technologie

Studijní obor:

2612T025 Informatika a výpočetní technika

Téma:

Aplikace rozšířené reality pro OS Android
Application of Augmented Reality for OS Android

Jazyk vypracování:

čeština

Zásady pro vypracování:

Rozšířená realita (augmented reality) je označení používané pro reálný obraz světa doplněný počítačem vytvořenými objekty. Jde o zobrazení reality (např. budovy nasnímané fotoaparátem mobilního telefonu) a následné přidání digitálních prvků nebo modelů do obrazu. Cílem této diplomové práce bude navrhnout aplikaci demonstrující možnosti rozšířené reality v prostředí OS Android. Demonstrační aplikací bude varianta hry Bomberman, odehrávající se na prostoru vymezeném značkami. Mobilní telefon detekuje značky položené na zemi a do tohoto obrazu vykreslí celé herní prostředí s 3D modely a pohybujícími se animovanými postavami.

1. Popis mechanismů a technik užívaných pro vytváření rozšířené reality.
2. Rešerše knihoven a aplikací pro práci s rozšířenou realitou.
3. Vytvoření animovaných 3D modelů postav a herního světa.
4. Implementace aplikace pro OS Android s využitím rozšířené reality.
5. Testování aplikace v různých podmínkách, popis typických problémových situací pro rozpoznávání.
6. Shrnutí dosažených výsledků (s možností publikování na Google Play).

Seznam doporučené odborné literatury:

- [1] Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, Prentice Hall; 3 edition, 2007, ISBN 978-0131687288
- [2] Alain V. Craig, Understanding Augmented Reality: Concepts and Applications, Morgan Kaufmann, 2013, ISBN 978-0240824086
- [3] Reto Meier, Professional Android 4 Application Development, Wrox, 3 edition, 2012, ISBN 978-1118102275

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Mgr. Ing. Michal Krumnikl, Ph.D.**

Datum zadání: 01.09.2014

Datum odevzdání: 29.04.2016



doc. Dr. Ing. Eduard Sojka
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární
prameny a publikace, ze kterých jsem čerpal.

V Ostravě 30. června 2016

.....
Kau J

Poděkování mám vcelku jednoduché. Nejvíce bych chtěl poděkovat vedoucímu práce Michalu Krumníkovi za velmi pěkný nápad na téma práce. Už uběhnul nějaký čas od doby, kdy jsem práci začal vypracovávat, ale určitě to byla výborná volba zajímat se více o rozšířenou realitu. Před dvěma lety jsem byl skeptický, jestli tyto vědomosti v brzké době využiji, a teď vidím, že poptávka je obrovská a i kolegové v práci se začínají o rozšířenou realitu více zajímat a přicházejí s jednoduchými projekty.

Abstrakt

Masové využití rozšířené reality se stalo aktuální až s příchodem výkonnějších mobilních zařízení. Začínají se objevovat první mobilní aplikace a rozšířená realita (AR) má velký potenciál být důležitou technologií v následujících letech. Proto může být přínosné se tímto tématem zabývat. Tato práce se věnuje především využití AR ve hrách a konkrétně ve hře Bomberman. Cílem práce by mělo být motivovat a inspirovat k vytvoření vlastní aplikace nebo alespoň objasnit, čím se AR zabývá a jaká jsou dnes možná využití. K tomu může pomoci rozbor dnes používaných AR knihoven, a poté nastínění principu jak některé z nich fungují. Dále pak popisují, které nástroje jsem k vývoji použil a které se mohou hodit i vám. Také popisují, jaké aplikace je možné sehnat již dnes.

Klíčová slova: rozšířená realita, hra, android, Bomberman

Abstract

Massive usage of augmented reality became actual, when more powerful mobile devices arrived. First AR mobile applications are arriving and augmented reality (AR) has big potential to be important technology in next years. That's why it can be useful to be interested in this topic. This work focuses primarily on the use of AR in games and specifically in the game Bomberman. The aim of the work should be to motivate and inspire you to create your own application or at least clarify what the AR means and for what may be used. To achieve this can help analysis of popular AR libraries and then light explanation how some of them work. Further I explain, which tools I used for development and some of them can be useful even for you. I also write about applications, which can be used even today.

Key Words: augmented reality, game, android, Bomberman

Obsah

Seznam použitých zkratk a symbolů	9
Seznam obrázků	10
Seznam tabulek	11
1 Úvod	13
2 Rozšířená realita	14
2.1 Vysvětlení pojmu	15
2.2 Typy AR	15
3 Možnosti AR her	18
3.1 HoloLens	18
3.2 Leap Motion	20
3.3 HTC Vive	20
3.4 Oculus Rift HD	21
3.5 Lenovo Phab 2 Pro - Projekt Tango	21
3.6 Cardboard	22
3.7 Mobilní zařízení	22
4 Problémy s rozpoznáním	24
4.1 Vyhledání features v obraze	24
4.2 Vhodný podklad	24
4.3 Světelné podmínky	24
4.4 Stabilní kamera	25
4.5 Objekt v záběru	25
4.6 Pevný podklad	25
5 Algoritmy pro rozpoznání	26
5.1 Rozpoznání pomocí markerů	26
5.2 Algoritmus pro rozpoznání markerů	26
5.3 Rozpoznáním zájmových bodů (features) v objektu	28
5.4 SIFT - Scale-Invariant Feature Transform	28
5.5 SLAM trackování	35
6 SDK pro práci s AR	36
6.1 Argon	36
6.2 ARToolKit	37

6.3	Mixare	37
6.4	Kudan	37
6.5	Vuforia	38
6.6	Wikitude	38
7	Návrh a Implementace hry	39
7.1	Inspirace	39
7.2	Volba použitých technologií	40
7.3	Změny oproti původní Dyna Blaster verzi	40
7.4	Knihovna Vuforia	41
7.5	Vytvoření herních modelů	43
7.6	Hra Labyrint	50
8	Závěr	52
	Literatura	53
	Přílohy	54

Seznam použitých zkratek a symbolů

AR	– Augmented reality
VR	– Virtual reality
HUD	– Head-up display
LoG	– The Laplacian of Gaussian
DoG	– The Difference of Gaussian
SLAM	– Simultaneous localization and mapping

Seznam obrázků

1	Rozdělení trhu s AR a VR	14
2	Navigace k bankomatům (Převzato z [10])	16
3	Mobilní klávesnice se promítá na prsty (Převzato z [11])	16
4	Z dnešního pohledu možná přehnané video ztvárnění AR. Doporučuji podívat se na celé video. (Převzato z [25]))	17
5	Ideální vzdálenost pro pozorování objektů s HoloLens (Převzato z [17])	19
6	Ukázka tvorby hologramu pomocí HoloLens (Převzato z [16])	19
7	Hraní hry Jenga (dřevěná věž) s pomocí Leap Motion (Převzato z [19])	20
8	Hra Longbow je součástí her „The lab“ od Valve (Převzato z [22])	21
9	Ukázka hry Woorld od Funomena (Převzato z [21])	22
10	Fibrum VR Roller Coaster Attraction (Převzato z [23])	23
11	Zombies Everywhere! (Převzato z [24])	23
12	Gaussova derivace (Převzato z [2])	27
13	Odfiltrování čar hran bez rohů (Převzato z [2])	27
14	Orientace hran (Převzato z [2])	28
15	DoG scale space pyramida pro SIFT (Převzato z [4])	30
16	Nalezení klíčového bodu (Převzato z [4])	30
17	Umístění extrému v mřížce.	31
18	Hodnoty a orientace gradientů (Převzato z [13])	33
19	Histogram orientací	33
20	Deskriptor klíčového bodu	34
21	SIFT klíčové body (Převzato z [15])	34
22	Bomberman AR	39
23	Data flow diagram - Vuforia SDK	43
24	Blender	44
25	Využití jedné textury pro více barev	45
26	Vývojářská část portálu Vuforia - nahrání „ <i>trackables</i> “	46
27	Ukázka ze hry Bomberman, promítnutá na českou bankovku.	47
28	Třídní diagram pro hru Bomberman	48
29	Zachycení joysticku a tlačítka pro pokládání bomb a jejich zvýraznění.	49
30	Obcházení zdí.	49
31	Labyrint - souřadnice scény a telefonu	51

Seznam tabulek

1	Rychlost FPS při různém nastavení aplikace.	50
---	---	----

Seznam výpisů zdrojového kódu

1	Část kódu pro lepší orientaci v ukázkových aplikacích Vuforia.	42
2	Získání projekční matice pro rozpoznané „ <i>trackables</i> “ (plochy) s pomocí knihovny Vuforia	44
3	Část kódu zjišťující natočení telefonu	50
4	Část kódu zjišťující natočení projekční matice vůči telefonu.	51

1 Úvod

Tato práce se věnuje zajímavému tématu rozšířená realita a snaží se najít a objasnit využití této relativně nové technologie. Poslední dobou se o rozšířené realitě mluví stále častěji. Využití na současných zařízeních je sice stále na hraně použitelnosti, ale pozvolna se začíná přicházet na to, kde všude by se dala rozšířená realita využít.

Pro většinu lidí je AR (rozšířená realita) neznámá věc a nic podobného zatím neviděli. Toho spousta vývojářů využívá a snaží se pomocí AR lidi oslovit méně tradičními aplikacemi nebo obsadit volné místo na trhu. Tento rok bude možná zlomový, protože přijde několik masově dostupných zařízení posouvajících možnosti rozšířené reality dále. Během dvou let si myslím povědomí o rozšířené realitě velmi vzroste a bude jen na vývojářích jestli se této nové možnosti chopí. Rozšířená realita skrývá zřejmě větší potenciál než chytré telefony a časem se jí nepůjde vyhnout.

Začátek bude zřejmě podobný jako u telefonů a nejprve budou vznikat velice jednoduché aplikace. Podobně jako se ale dnes aplikace na mobilních zařízeních přibližují těm desktopovým, tak i AR aplikace se stanou brzy velice komplexními. Když jsem začal psát tuto práci, tak jsem sotva věděl něco o Google Glass a dnes se jedná již o zastaralý produkt. Podobně zřejmě dopadnou HoloLens a několik dalších zařízení, než se nalezne správná technologie pro podobný typ displeje a pro ovládání.

Na úvod bych měl asi nastínit o čem rozšířená realita je, i když se to podrobněji dočtete v další kapitole nebo AR již znáte. Uvedu proto jednoduchý příklad. Dnes ve městech máme poměrně rozšířené billboardy a plakáty. Ve městě budoucnosti s AR již billboardy nebudou, ale reklamy přesto nebudou a budou k nám vystupovat v animovaných 3D videích téměř ze všeho. Uvidíme je zřejmě přes brýle a bude možné s nimi interagovat. Možná to je skoro až odstrašující příklad, ale je třeba se na to podívat s nadhledem a přiznat si, že AR bude mít v reklamě veliké využití (obr. 4).

Hlavním cílem této práce, jak už je ze zadání patrné, je zaměření na hry, konkrétně hru Bomberman s použitím OS Android. Při podrobnějším rozboru her s AR pro Android se brzy nevyhneme zkoumání konkurenčních platforem, abychom si udělali ucelenější obrázek současného stavu AR. O dnešních zařízeních a některých jejich využitích se dočtete v šesté kapitole.

V dalších kapitolách se dočtete, jaké mohou nastat problémy při rozpoznávání nebo jaké algoritmy je možné k rozpoznávání použít. V kapitole o návrhu a implementaci popisují jak jsem aplikaci vytvořil, vlastnosti zvolené knihovny pro rozšířenou realitu a ukazují jak ji použít pro vytvoření AR aplikace.

Závěrem práce bude zhodnocení výsledné aplikace a dosažených výsledků.

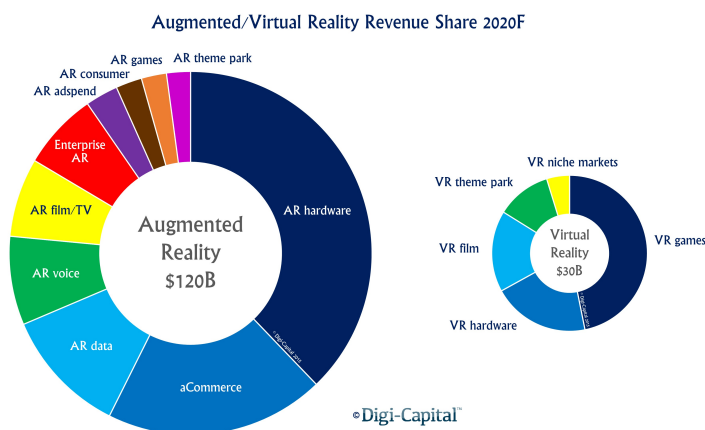
2 Rozšířená realita

Rozšířená realita má silné propojení s virtuální realitou. Obě vytvářejí 3D svět vytvořený počítačem. Virtuální realita se snaží o úplné nahrazení a vtažení člověka do „virtuálního“ světa a rozšířená realita se snaží o propojení našeho reálného světa se světem virtuálním.

Proč se rozšířenou realitou zabývat? Jedná se o dynamickou oblast a s příchodem výkonných mobilních zařízení se může stát, že nalezne i významnější reálné uplatnění v běžném životě. V dnešní internetové době vzniká spousta vynálezů, ale jen málokterý má potenciál přiblížit lidstvo ke sci-fi filmům jako právě rozšířená realita (AR).

Mnozí lidé, kteří se zabývají technologickými trendy, vidí budoucnost v mobilních zařízeních, internetu věcí, autonomních automobilech a neopomínají ani AR a VR.

Navíc i velké společnosti investují nemalé částky do AR. Facebook investoval 2 miliardy dolarů do Oculusu. Google vyvíjí Google Glass a investoval 542 miliónů dolarů za Magic Leap a Microsoft má velké plány s nadějně vyhlížejícími HoloLens. Nedávno Microsoft představil také tzv. „holoportaci“, která nevypadá vůbec špatně a má potenciál posunout mezilidskou komunikaci zase dále.



Obrázek 1: Rozdělení trhu s AR a VR v roce 2020 (Převzato z [9])

S AR se nabízí možnost mít televizi/monitor kdekoli nás napadne a videohovory s mobilem v ruce mohou být poměrně brzy minulostí, protože mobilní displej už nebude potřeba. Můžeme „mít“ namísto starého auta virtuální Ferrari nebo, kdo bude chtít, vesmírnou loď. Lidé mohou využívat znalostí Wikipedie jen při pohledu na určitý předmět. Zaměstnanci mohou být s AR lépe zaškoleni pro konkrétní úkoly ve své práci například pomocí animací v HUD (displej připevněný k hlavě) a hlasových návodů.

Využití je ohromné a je jen otázka času, kdy AR naplno změní naše životy. Ve velmi blízké budoucnosti se začnou také prosazovat implantáty. Ze začátku půjde jen o čipy pod kůží nebo

náhrady srdce a bude s tím souviset i spousta odmítavých názorů. Později nás ale zřejmě čeká svět s očními implantáty nebo i těmi mozkovými, a to bude ještě mnohem větší význam pro rozvoj AR a VR.

2.1 Vysvětlení pojmu

Jako první použil pojem „rozšířená realita“ Tom Caudell v roce 1992 [7], ale nápad jako takový se objevil již dříve.

Pojem lze definovat jako technologie, které nám umožňují rozšířit okolní svět ve kterém žijeme o virtuální prvky.

V praxi si jako příklad AR můžeme představit deskovou hru, kde se hráčům na prázdném stole promítají figurky, a ti je mohou ovládat jako by tam skutečně byly, ale to je jen nepatrný zlomek využití. AR může lidem usnadnit a rozšířit komunikaci s počítači nebo ji udělat zábavnější. Představit si můžeme také využití v reklamě (obr. 4).

2.2 Typy AR

Ten kdo se dnes již s AR setkal, může snadno nabýt dojmu, že jde o 3D objekty, které mu doplňují jeho reálný svět. AR ale není omezená pouze na zrak a existují aplikace, kde se zrak vůbec nevyužívá. Může se zdát rozumné rozdělit AR podle lidských smyslů. Do jisté míry to není špatný nápad, jen je třeba zohlednit, že zrak bude v dnešní době mít největší využití a pro chuť asi nebude snadné najít větší uplatnění. Mnohé aplikace mimozrakových způsobů AR si můžeme snadněji představit, když se pokusíme vžít do role nevidomého člověka.

2.2.1 Pomocí lokace

Tento typ AR využívá dnes nejčastěji GPS, akcelerometr a kompas. Uživatel se pak mohou ukázat na zařízení (Google Glass, mobilní telefon, HoloLens...) jeho přátele v okolí, bankomaty nebo se dá použít i ve hrách. Úspěšnou hrou v tomto žánru je například *Zombie run*. Ten je navíc specifický i tím, že k navození atmosféry používá výhradně zvuk a tím mezi AR programy výrazně vybočuje a pomáhá nahlížet na AR aplikace i jinak.

2.2.2 Projekcí

O AR projekci se také často hovoří, i když není tolik známá. Jeden typ využití může být interakce s uživatelem, kde jako na obrázku použije svou ruku jako mobilní klávesnici. Jiné využití může být „holografické“, kdy z nějakého důvodu není vhodné mít HUD - *Head-up display*. Například v budoucnu se může část těla pacienta promítnout na vedlejší virtuální sál, ve kterém budou pracovat lékaři, a reálnou operaci provedou roboti ve vedlejším sále, aby si nepřekáželi nebo celá operace proběhne uvnitř pacienta, do kterého by lékaři stejně bez větších řezů neviděli. Tato AR ale bude zřejmě mít příliš silnou konkurenci v HUD, takže bych si na větší rozšíření příliš nevsázel.



Obrázek 2: Navigace k bankomatům (Převzato z [10])



Obrázek 3: Mobilní klávesnice se promítá na prsty (Převzato z [11])

2.2.3 Hmatem

V dnešní době je možné použít ultrazvuk, který na hladkých displejích dává člověku pocit zpětné hmatové odezvy. Ta může být v podobě neviditelného tlačítka nebo se také často prezentuje odhrnutí zrněk písku z displeje. Taková zařízení mají zatím pro mobilní využití velkou spotřebu a jsou lehce hlučná. I přesto se s nimi zřejmě brzy setkáme, protože mohou velmi pomoci zrakově postiženým, ale i ostatním se může hodit zpětná odezva displeje, například při řízení automobilu. Mnohem lepší zážitek by mohla přinést hmatová odezva ve 3D [6], u ní by bylo značné využití i ve hrách, například v deskových.

2.2.4 Sluchem a hlasem

Ovládání hlasem si asi není těžké představit, mobilní telefony dnes ovládání hlasem umožňují a dávají zpětnou hlasovou odezvu. O tomto se často nemluví jako o AR, i když podle definice není příliš rozdíl, jestli se zeptám zařízení na nejbližší restaurace, a to mě bude navigovat hlasem nebo jestli stejnou akci provedu na dotykovém displeji a nechám se navigovat s Google Glass. Kde už se ale o sluchové AR nesporně mluví, je hra *Zombies, Run!*, která se snaží dostat běžce do světa, ve kterém je pronásledován nemrtvými a s pomocí zvuků, GPS a akcelerometru se zjišťuje jeho poloha, a jak běžec plní úkoly.

2.2.5 Zobrazením VR na displeji

Jde o nejširší skupinu, kterou si většina zřejmě oprávněně hned vybaví pokud se mluví o AR. Podstatou této skupiny, je zobrazit na displeji uživateli reálný svět. To je možné přes průhledný displej (např. HoloLens) nebo kamerou (např. mobilní telefon). Do tohoto světa pak je třeba zobrazit nějaké virtuální prvky. Může jít o text, video nebo 3D objekty v takové kvalitě, že je nebude možné rozeznat od těch reálných. Tento typ bychom mohli rozdělit do několika podskupin, které jsou omezeny technologickým pokrokem. Dnes se jedná pouze o mobilní telefony a tablety, které má doma téměř každý. Časem přijdou zařízení v podobě brýlí a to, co bude následovat, už zřejmě nepůjde nazývat displejem. Způsoby zobrazení se relativně bouřlivě vyvíjejí. V praxi by pak mohly vznikat aplikace, které mohou například v nové firmě všem zaměstnancům na krk zobrazit virtuální cedulku s jejich jménem. Někdo jiný by mohl chtít vidět namísto aut na ulici trabanty nebo by všechna auta měla stejnou barvu podle výrobce nebo nějakého jiného kritéria. Možnosti jsou nekonečné.



Obrázek 4: Z dnešního pohledu možná přehnané video ztvárnění AR. Doporučuji podívat se na celé video. (Převzato z [25])

3 Možnosti AR her

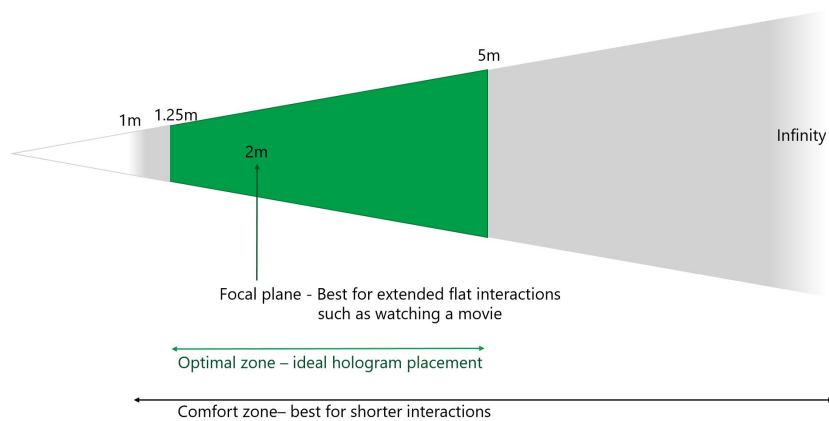
Pokud si dnes chceme užít AR doma, můžeme si vybrat z několika možností. Vybral jsem několik masově rozšířených zařízení nebo zařízení s potenciálem se rozšířit. Ke každému jsem se pokusil nalézt vhodné ukázky her a využití. Uvedené ceny jsou aktuální k 1.7.2016. Jedná se o poměrně různorodá zařízení, pokud porovnáme Leap Motion a HoloLens, ale každé cílí na jinou skupinu uživatelů a cenově jsou také naprosto odlišné. Není snadné nalézt ostrou hranici mezi tím, co lze ještě považovat za AR a co již ne. Například Duck Hunt od Nintenda z roku 1985 používal k ovládání pistoli, kterou hráč mířil na televizní obrazovku. Když se podíváme na definici AR ve druhé kapitole, tak o Duck Hunt můžeme pochybovat, jestli by šlo zařadit do AR. U Leap Motion, kde používáme vlastní části těla jako ovládání, už je myslím si jasnější, že se jedná o nový způsob ovládání pro AR zařízení, který je o něco dál než touchpad nebo klávesnice.

Pokud se zaměřím pouze na herní využití, pak jednoznačně HTC Vive a Oculus nemají konkurenci. Za nimi by byl zřejmě Cardboard, který dominuje množstvím her. Pro ostatní zařízení zatím příliš her neexistuje, mají ovšem potenciál i mimo hraní her.

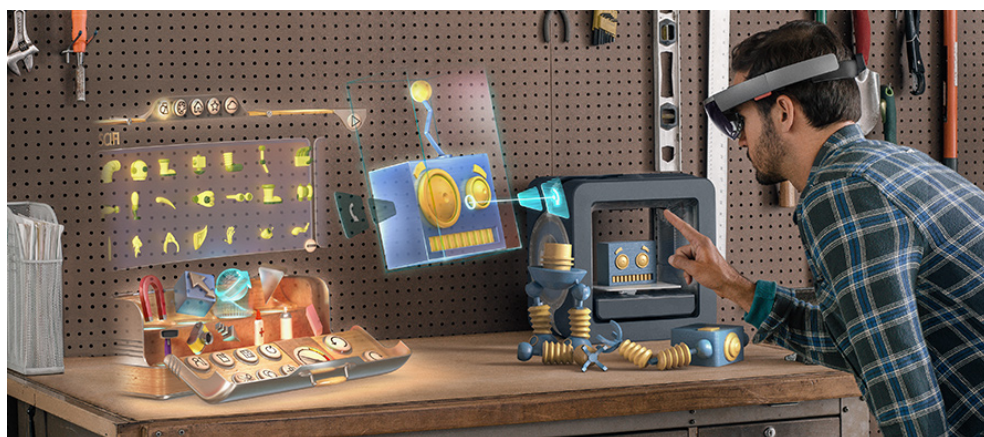
3.1 HoloLens

- Vydání: **30. března, 2016**
- Přibližná cena: **75 000 Kč + DPH + poplatky (\$3 000 USA)**

Je to dnes zřejmě nejpokročilejší zařízení pro AR. V České Republice je zatím obtížně dostupné. Největším mínusem je nedostatek her a aplikací. Podle recenzí jde stále o zařízení spíše pro nadšence než pro masu. Měl jsem možnost HoloLens otestovat a nereagují dostatečně rychle a přesně. Monitor by nahradit do jisté míry zvládly, ale k tomu aby dokázaly nahradit myš a klávesnici mají ještě hodně daleko. Podle mých zkušeností probíhá většina uživatelských vstupů přes kurzor, který je umístěn ve středu obrazovky. Uživatel pak může nejčastěji zmáčknutím palce a ukazováčku provést nějakou akci s tímto kurzorem. Tím může uchopit předmět a ten rukou přesunout nebo stisknout nějaké tlačítko, které může být spojeno s vybraným předmětem. Silná stránka HoloLens je určitě v reálném rozpoznávání okolí a tvarů v něm. Kvalita displeje je diskutabilní, nabízí rozlišení 1268 x 720 pro každé z očí. Problém je také se vzdáleností, kde se *hologramy* mohou zobrazit (obr. 5). V porovnání s Google Glass je displej o kus větší a je praktičtěji umístěn doprostřed. Velikost obrazovky se uvádí jako 60 cm vzdálená plocha s úhlopříčkou 38 cm a to je stále málo.



Obrázek 5: Ideální vzdálenost pro pozorování objektů s HoloLens (Převzato z [17])

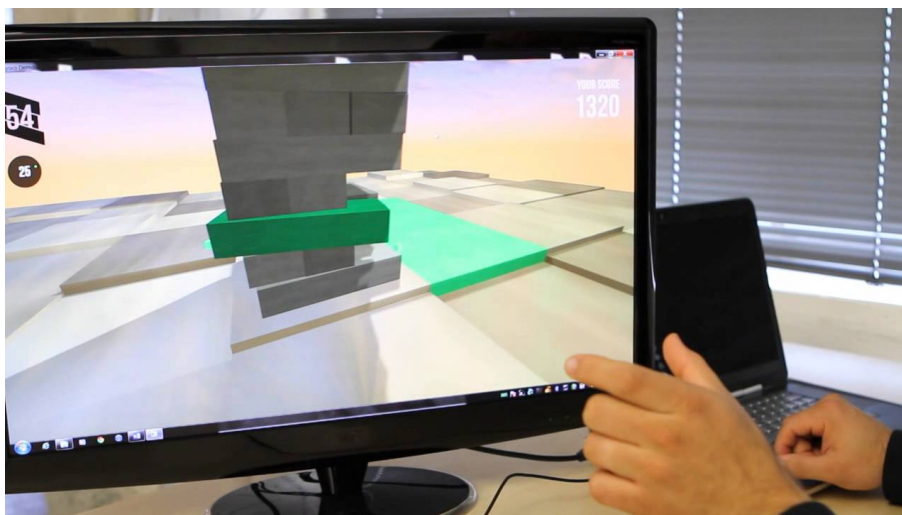


Obrázek 6: Ukázka tvorby hologramu pomocí HoloLens (Převzato z [16])

3.2 Leap Motion

- Vydání: **15. května, 2013**
- Přibližná cena: **2 400 Kč**

Leap Motion jako jediné z výše uvedených nemá vlastní displej. Výhodu ale může mít v rozšíření i napříč AR/VR zařízeními. Nabízí zřejmě nejpřesnější detekci rukou, a ta je pro používání AR jedním z klíčových prvků. Leap Motion spolupracuje také s velkými výrobci notebooků jako je HP nebo Asus, kteří jej do některých svých notebooků instalují. V únoru 2016 vydali nový software speciálně určený pro použití ve VR - Orion. Leap Motion je kompatibilní s HTC Vive i Oculus Rift HD. Leap Motion má také potenciál nahradit klávesnici psaním gesty, které lze využít i na mobilním telefonu. Bohužel zatím existuje zřejmě jen projekt Minuum [20], který ale příliš nevyužívá možnosti Leap Motion a neumožňuje ani psaní poslepu, které se u psaní gesty velmi nabízí.



Obrázek 7: Hraní hry Jenga (dřevěná věž) s pomocí Leap Motion (Převzato z [19])

3.3 HTC Vive

- Vydání: **5. dubna, 2016**
- Přibližná cena: **24 000 Kč**

Jedná se o headset se dvěma ovladači. Neumí zdaleka interagovat s okolím jako výše uvedené HoloLens, ale i přes to budou HoloLens konkurovat nebo se s nimi porovnávat, pokud půjde o hry. HTC Vive má přední kameru, která umožňuje rozpoznat okolí, ale neumí rozpoznávat hloubku a slouží zřejmě jen na základní orientaci a ve hrách zřejmě bude použita minimálně. Hráč se může pohybovat v prostoru asi 4,5 x 4,5 metrů



Obrázek 8: Hra Longbow je součástí her „The lab“ od Valve (Převzato z [22])

3.4 Oculus Rift HD

- Očekávané vydání: **září, 2016**
- Přibližná cena: **17 500 Kč**

V mnohém je srovnatelný s HTC Vive. Má shodně jako HTC rozlišení pro každé oko 1080 x 1200 při 90Hz obnovovací frekvenci. Nemá přední kameru a nedokáže tedy reagovat na okolí. Hráč se může pohybovat v prostoru 3,3 x 1,5 metrů.

3.5 Lenovo Phab 2 Pro - Projekt Tango

- Vydání: **září, 2016**
- Přibližná cena: **17 000 Kč + DPH + poplatky (\$499 USA)**

Jde o první mobilní telefon s technologií Tango. Tango je v mnoha ohledech podobné HoloLens. Snaží se také o rozpoznání okolí a o masovější rozšíření AR, ovšem Google na to jde odlišně. Tango chtějí rozšířit přes cenově dostupné telefony a tablety. Sice to není tak praktické jako HUD, ale tady je důležitá cena a málokdo bude moci investovat řádově desetinásobné částky do HoloLens, které stejně brzy zastarají.

V ukázkách se často počítá s tím, že uživatel bude mít naskenovaný byt, a pak se v obchodě bude moci zeptat „Ok Google, jak vysoké mám doma žaluzie?“ Nebo při nákupu nábytku si bude moci rovnou prohlédnout, jak nový kousek zapadne do místnosti. Další veliké využití bude v navigacích uvnitř budov, protože zařízení bude schopno okamžitě rozpoznat polohu v naskenovaném objektu a také svou přesnou orientaci v prostoru. A pro velkou část populace

bude důležité také využití ve hrách. Zajímavě vypadá hra Woorld (obr. 9), ve které je ale úroveň zábavnosti spekulativní a jde spíše o demonstraci možností.



Obrázek 9: Ukázka hry Woorld od Funomena (Převzato z [21])

3.6 Cardboard

- Vydání: **25. června, 2014**
- Přibližná cena: **150+ Kč**

Pro první seznámení s AR může být cardboard nejlepší volba. Výhodou je nízká pořizovací cena, protože je potřeba mít pouze vlastní mobilní telefon. Pokud ale chceme mít alespoň trochu přijatelný obraz, je nutnost mít mobilní zařízení s rozlišením 4K a více. (obr. 10). Velká většina graficky povedených her pro Cardboard vznikla v Unity IDE ¹.

3.7 Mobilní zařízení

- Vydání: ?
- Přibližná cena: ?

Jedná se pouze o obyčejný telefon nebo tablet, takže cenu je obtížné stanovit, a také určit, který telefon byl první vhodný pro AR asi nedává moc smysl. Pro tento typ jsem se rozhodl vytvářet hru v rámci této práce. Výhodou je určitě rozšíření. Nevýhodou může být ohromná fragmentace zařízení a slabší AR zážitek. (obr. 11) Pro mobilní telefon není snadné vymyslet aplikaci, která bude využívat AR a bude mít nějakou přidanou hodnotu. Taky mobilní hry s AR na Google Play rozhodně nelámou rekordy a pokud si ji někdo stáhne, tak je to spíše ze zvědavosti.

¹<https://unity3D.com/>



Obrázek 10: Fibrum VR Roller Coaster Attraction (Převzato z [23])



Obrázek 11: Zombies Everywhere! (Převzato z [24])

4 Problémy s rozpoznáním

V této kapitole popisují na jaké problémy nejčastěji narazíme pokud budeme chtít nalézt kamerou pozici nějakého objektu. Nejedná se o žádné neočekávané problémy a závisí také na konkrétním algoritmu a vlastnostech fotoaparátu.

4.1 Vyhledání features v obraze

Stabilita obrazu a celkově rozpoznání závisí na dostatečném počtu nalezených bodů vhodných k trackování. Pokud se nepodaří najít dostatečné množství bodů, pak bývá objekt roztřepený.

4.2 Vhodný podklad

Nevhodně zvolený podklad udělá nefunkční celou aplikaci nebo alespoň uživatelský zážitek z ní bude špatný. Naopak u vhodně zvoleného mohou uživatelé skoro až žasnout, když je možné podklad rozeznat v krajním přiblížení, nebo když se téměř vytrácí ze záběru. U podkladu bychom měli klást důraz na následující:

- **Bohatý na detaily:** správný obrázek použitý k trackování by neměl obsahovat například zpola zaplněný stůl, ale detaily musí být rovnoměrně rozmístěné v celém obrázku. Vhodným podkladem může být například kamenitá cesta, která ale musí být dobře zaostřená a dostatečně přiblížená, nebo spadané listí.
- **Bez opakujících se vzorů:** tím jak funguje vyhledávání *feature points*, jelikož se při vyhledávání *feature points* hledá vzor zastoupený v databázi, nesmí algoritmus najít stejný vzor na více místech. Například v případě kamenité cesty zmíněné výše, je třeba dbát na to, aby obrázek neobsahoval hromadu hladkých podobně tvarovaných a osvětlených kamenů. Další možnou špatnou volbou může být písmo. To je sice kontrastní a s dostatečným množstvím detailů, ale písmena se často opakují.
- **Dostatečný kontrast:** mnoho algoritmů pro vyhledávání feature bodů pracuje s černobílým obrazem, takže se může stát, že zcela ignorují barevný kontrast. Pro vyhledání extrémů, mezi kterými se hledají feature body, je ale potřeba určitého kontrastu dosáhnout.

4.3 Světelné podmínky

Pro dobré rozpoznání by měl trackovaný obrázek být rovnoměrně a dostatečně osvětlený. Neměly by do něj zasahovat stíny okolních předmětů, ani by neměl být osvětlovaný ze strany bodovým světlem. Toho může být těžké dosáhnout u sledování 3D objektu.

4.4 Stabilní kamera

Při prudším pohybu dojde k rozostření a na několik sekund nemusí být obraz možné rozpoznat. K tomu již nebude stačit využití feature bodů, protože ty v rozmazaném obraze lze velmi obtížně nalézt. Je možné použít akcelerometr, který takový prudký pohyb zaznamená a bude trackovaný objekt očekávat na předchozím místě, nebo jinou techniku rozpoznání obrazu. Akcelerometr ale nepomůže, pokud se prudce pohne sledovaný objekt.

4.5 Objekt v záběru

Pozici objektu je možné předpokládat a s pomocí gyroskopu a akcelerometru je možné ji udržet, i pokud se objekt dostane mimo kameru. S tímto do jisté míry pracuje Kudan SDK.

4.6 Pevný podklad

Rozpoznání hodně neprospívá použití měkkého materiálu pro trackovaný podklad. Na měkkém papíru budeme mít velké problémy a scéna bude roztřesená. Pokud jej podlepíme dosáhneme výrazně stabilnějších výsledků.

5 Algoritmy pro rozpoznání

V této práci se zaměřuji výhradně na oblast rozšířené reality, která se věnuje zobrazení virtuálních 3D modelů na obraze z kamery. Zásadní částí je nalezení podkladu, na kterém se má většinou 3D objekt vykreslit. Pro tento účel se často používá algoritmus SIFT nebo SURF, které dokážou v reálném čase nalézt body v obraze, nezávisle na jeho velikosti nebo natočení.

5.1 Rozpoznání pomocí markerů

Jedná se o vyhledání předem očekávaných značek - markerů. Ty si reálně můžeme představit jako bílé kartičky s černými rámečky. Ty uvnitř běžně obsahují dodatečnou informaci např. v podobě QR kódu nebo písmene pro identifikaci konkrétního markeru, pokud jich je více. Vzhled markerů se volí tak, aby je náš algoritmus mohl snadno rozpoznat v obraze. Výhodou použití markerů oproti univerzálnějším metodám hledajících *features body*, je menší výpočetní náročnost. Tím je možné využít výkon pro více markerů nebo lepší přesnost/stabilitu rozpoznání.

5.2 Algoritmus pro rozpoznání markerů

Tento algoritmus implementoval Corné [2], ale existuje více způsobů rozpoznání markerů.

V prvním kroku obrázek rozdělíme do menších regionů 40x40 pixelů a pak i tyto regiony do mřížky 8x8 jako na obrázku 12. Další 3 kroky se provedou již pouze uvnitř těchto malých regionů, a tím se celý algoritmus velmi urychlí.

Pro pixely ležící na mřížce uvnitř regionů aplikujeme konvoluční matici pro zvýraznění hran (1). Pokud po provedení filtru překročíme stanovenou hodnotu, označíme tento pixel pro další výpočet.

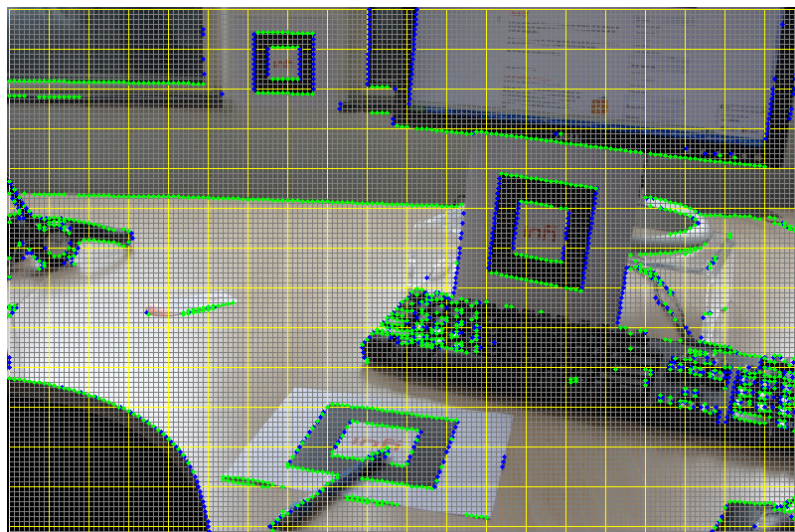
$$\begin{bmatrix} -3 & -5 & 0 & 5 & 3 \end{bmatrix} * \mathbf{A} \quad (1)$$

Pro každý takto označený pixel rozhodneme pomocí Sobelova operátoru (2) o jeho orientaci. Horizontálně orientované označíme zeleně a vertikálně orientované modře. Dostaneme pak výstup podobný obrázku 12.

$$\mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A}, \quad \mathbf{G}_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * \mathbf{B} \quad (2)$$

Pomocí RANSAC (RANdom SAMple Consensus) algoritmu zjistíme orientaci hran.

Algoritmus funguje tak, že náhodně vybere 2 body ve stejném regionu a se stejnou orientací, které mohou teoreticky vytvořit hranu. Kolem této imaginární hrany se spočítají body, které



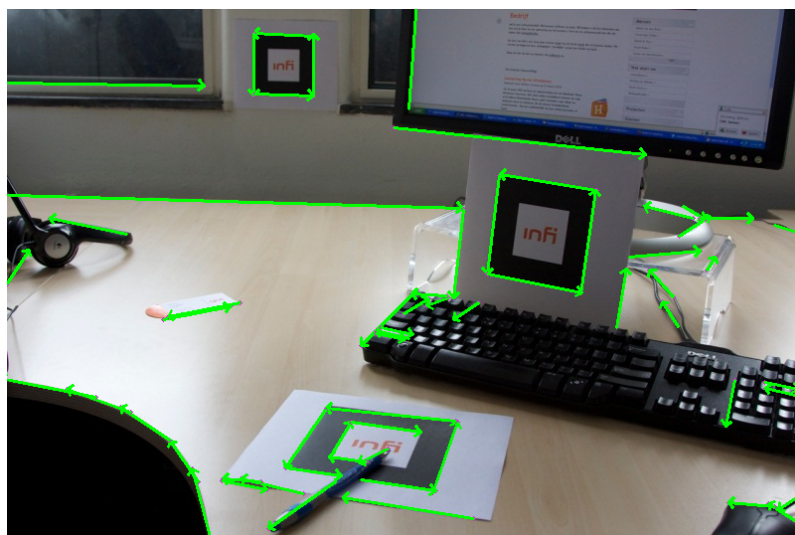
Obrázek 12: Gaussova derivace (Převzato z [2])

mají stejnou orientaci jako 2 původní. Tyto kroky se provedou přibližně 25 krát a všechny hrany, které mají alespoň 4 body můžeme považovat za hranu pro další výpočet.

Nyní jednotlivé hrany spojíme se sousedními z vedlejších regionů. Kontrolujeme zda jsou konce hran blízko u sebe a se stejnou orientací.

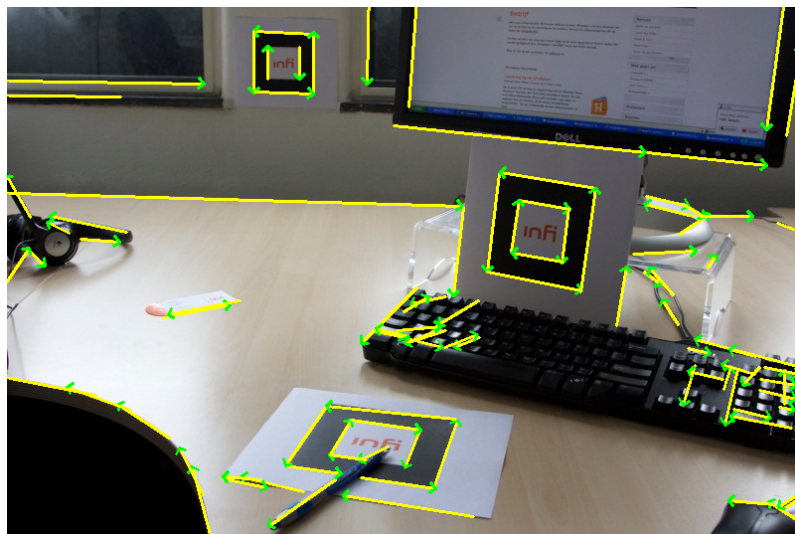
Takto získané hrany protáhneme, aby byla detekce přesnější. Hrany prodlužujeme na obě strany pixel po pixelu, dokud konvoluční matice (1) nenarazí na konec hrany.

V dalším kroku se odfiltrují hrany bez rohů (obr. 13).



Obrázek 13: Odfiltrování čar hran bez rohů (Převzato z [2])

Při finálním rozpoznání markerů se hledají 3-4 spojené hrany. Z nich vytvořené obrázce budou na pokrajích směrem dovnitř obsahovat černou.



Obrázek 14: Orientace hran (Převzato z [2])

5.3 Rozpoznáním zájmových bodů (features) v objektu

Zřejmě i v budoucnu nejrozšířenějším druhem AR bude promítání virtuálních prvků na rozpoznaný obraz popř. 3D tvar. První nelehká část je rozpoznat objekt, o to se snaží velká spousta lidí. Aktuálně jsou pro mobilní využití vhodné metody vyhledávání tzv. *zájmových bodů* (*feature points*) nebo je možné použít předpřipravené markery. Dá se očekávat, že algoritmy půjdou ještě dále. Pokud se podaří využít algoritmy z autonomních aut, bude možné rozpoznat mnohem více i v relativně neznámém prostředí. Později se dá očekávat zapojení vyvinutější umělé inteligence.

Pro rozpoznání zájmových bodů lze použít více algoritmů. Mezi ty nejznámější patří SIFT, který je popsán v následující kapitole 5.4.

5.4 SIFT - Scale-Invariant Feature Transform

Algoritmus SIFT se snaží hledat v obraze lokální zájmové body definované svými vektory. Ty jsou v ideálním případě vzájemně invariantní k rotacím nebo změnám velikostí v prohledávaném obraze. Je považován za poměrně robustní algoritmus. SIFT dokáže s vysokou pravděpodobností nalézt hledaný obraz ze tří klíčových bodů. Dokáže sledovat několik obrázků vůči rozsáhlejší databázi. V původní verzi [4] se používal pro černobílé obrázky, ale je možné jej upravit i pro barevné rozpoznávání. Vzniklo také mnoho dalších rozšíření, se kterými je možné sledovat plochy u 3D objektů nebo rozeznávat tváře.

Algoritmus se skládá ze čtyř částí:

5.4.1 Scale-Space Extrema Detection

Jedním z prvních algoritmů používaných pro nalezení *blobů* (výrazných skvrn v obrázku) je *Laplacian of Gaussian - LoG*.

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \quad (3)$$

Stejných výsledků lze ale také dosáhnout s *Difference of Gaussian - DoG*, který je výpočetně méně náročný. Lze jej spočítat jako limitní případ dvou Gaussově rozostřených obrázků (5). Vytvoříme pomocí ní „rozdíl Gaussových pyramid“. Není pevně dáno kolik *oktáv* a stupňů rozostření se má použít. Nicméně Lowe [4] odzkoušel a doporučil pro svůj algoritmus použít 4 *oktávy* a 5 stupňů rozostření (obr. 15). První *oktáva* se skládá z 5 větších čtvercových obrázků, u kterých se aplikuje Gaussovo rozostření. Pro každou další *oktávu* se počáteční hodnota σ zdvojnásobuje. Velikost obrázků v každé vyšší *oktávě* je poloviční.

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (4)$$

L - rozostřený obrázek

G - Gaussov operátor rozostření

I - obrázek

x, y - souřadnice

σ - parametr pro zvětšení, rozostření

$*$ - je konvoluční operátor

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \quad (5)$$

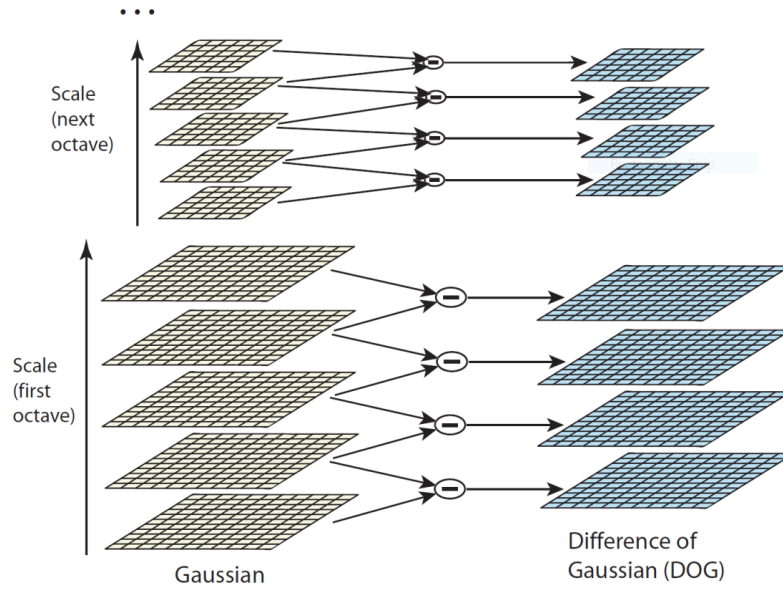
Nyní projdeme všechny pixely ve výsledných DoG obrázcích a porovnáme je se všemi sousedními pixely. Porovnání probíhá ve 3D, tedy v oblasti 3x3 pixely. Pokud máme v každé oktávě po 5 obrázcích, po provedení DoG máme výsledné 4. Okrajové obrázky nemáme s čím porovnat, proto nám vzniknou pouze 2 obrázky s možnými klíčovými body (obr. 16). X je zvoleno za kandidáta pro klíčový bod, pokud je větší nebo menší než všech 26 okolních bodů.

5.4.2 Nalezení klíčových bodů - keypoints

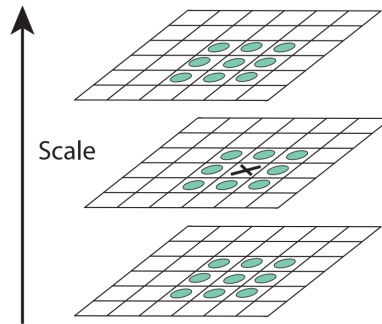
Nalezené body jsou zaokrouhlená maxima nebo minima, protože byly hledané do přesnosti pixelů a obrázky byly navíc zvětšovány. Klíčové body se také mohou nacházet velmi blízko u sebe.

5.4.2.1 Odstranění bodů s nízkým kontrastem

V dalším kroku se tedy pokusíme určit přesněji lokace a hodnoty extrémů (obr. 17). Z nich pak můžeme lépe vyloučit extrémy s malým kontrastem nebo ty, které jsou obtížně rozeznatelné v okolí hran. V původní verzi algoritmu se zaokrouhlení ignorovalo a použila se přímo



Obrázek 15: DoG scale space pyramida pro SIFT (Převzato z [4])



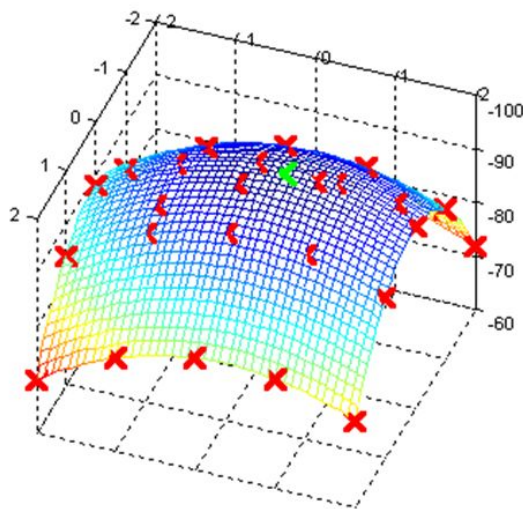
Obrázek 16: Nalezení klíčového bodu (Převzato z [4])

pozice a hodnota již nalezeného bodu. Zjistilo se, že po upřesnění dojde k lepšímu rozpoznání a stabilitě. Zpřesnění se dosáhne Taylorovým polynomem kvadratického stupně pro *scale-space* funkci $D(x, y, \sigma)$. Ta se posune tak, že její počátek umístíme do jednoho z dosud neupřesněných kandidátů na klíčové body:

$$D(\mathbf{x}) = D + \frac{\partial D^T}{\partial \mathbf{x}} \mathbf{x} + \frac{1}{2} \mathbf{x}^T \frac{\partial^2 D}{\partial \mathbf{x}^2} \mathbf{x} \quad (6)$$

kde D a její derivace jsou ohodnoceny pro aktuální bod a $\mathbf{x} = (x, y, \sigma)^T$ je odsazení od tohoto bodu. Pozici extrému $\hat{\mathbf{x}}$ spočítáme jako derivaci této funkce podle \mathbf{x} a dosazením 0 dostaneme

$$\hat{\mathbf{x}} = -\frac{\partial^2 D^{-1}}{\partial \mathbf{x}^2} \frac{\partial D}{\partial \mathbf{x}} \quad (7)$$



Obrázek 17: Umístění extrému v mřížce.
Extrém je znázorněn zeleně. (Převzato z [12])

Jestliže $\hat{\mathbf{x}}$ je větší než 0.5 ve kterékoliv dimenzi, pak to znamená, že extrém leží blíže jinému bodu.

Podle funkční hodnoty extrému $D(\hat{\mathbf{x}})$ můžeme rozhodnout o jeho slabém kontrastu a jeho nestabilitě. Substitucí (7) do (6) získáme vztah:

$$D(\hat{\mathbf{x}}) = D + \frac{1}{2} \frac{\partial D^T}{\partial \mathbf{x}} \hat{\mathbf{x}} \quad (8)$$

Jestliže hodnota $|D(\hat{\mathbf{x}})|$ nedosahuje prahové hodnoty 0.03, pak je tento klíčový bod vyloučen. Takový extrém je odebrán na základě nízkého kontrastu.

5.4.2.2 Odstranění hran

K nalezení kvalitních klíčových bodů, nám nestačí odstranit pouze body na základě nízkého kontrastu. DoG má totiž tendenci nalézat body v blízkosti hran, které nejsou příliš stabilní. Jako řešení tohoto problému se ukázalo spočítat dva gradienty v klíčovém bodě, oba navzájem kolmé [5]. V obraze kolem klíčového bodu, pak může být:

- **Rovina:** nastane, pokud jsou oba gradienty malé
- **Hrana:** jeden gradient bude malý (podél hrany) a druhý velký
- **Roh:** oba gradienty budou velké

Gradientsy spočítáme z Hessiany matice 2 x 2 (9). Nepotřebujeme počítat jednotkové vektory, ale stačí nám zjistit pouze jejich poměr.

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \quad (9)$$

Hodnota $(r+1)^2/r$ je v minimu, pokud jsou oba jednotkové vektory stejně dlouhé a narůstá s r . Abychom rozhodli, jestli je hodnota pod prahovou hranicí r , musíme pouze ověřit (10). Tento výpočet je výpočetně nenáročný. Za r se často dosazuje 10.

$$\frac{Tr(\mathbf{H})^2}{Det(\mathbf{H})} < \frac{(r+1)^2}{r} \quad (10)$$

5.4.3 Přiřazení orientace

Každému klíčovému bodu nyní přidělíme jednu nebo více orientací, podle lokálního směru gradientu v obrázku. Všechny budoucí výpočty prováděné nad daty v obrázku, které budou relativně transformované k dané orientaci, zvětšení a pozici pro každý klíčový bod, se pak tímto stanou nezávislými na těchto transformacích.

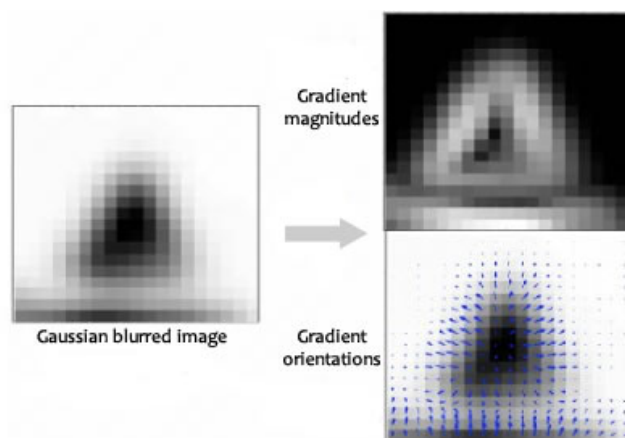
Každému klíčovému bodu přiřadíme orientaci na základě vlastností obrázku. Díky tomu můžeme deskriptor vyjádřit relativně k orientaci, a tím dosáhnout nezávislosti na rotaci. Nevýhodou tohoto řešení je omezení informací, které obsahuje deskriptor. Jsou z něj odstraněny informace, které nejsou nezávislé na orientaci.

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \quad (11)$$

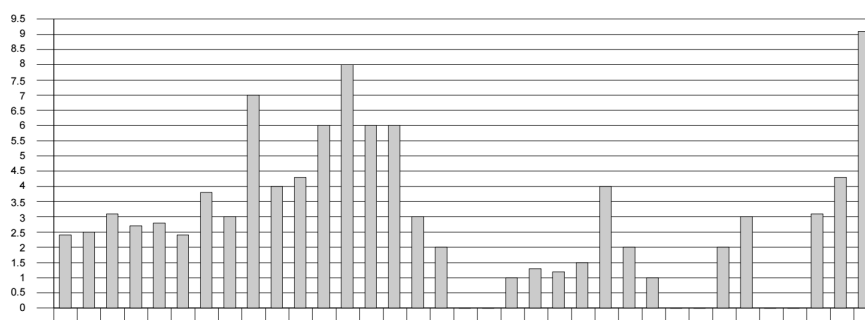
$$\theta(x, y) = \tan^{-1} \frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)} \quad (12)$$

Pro všechny okolní pixely spočítáme hodnotu (*magnitude*) (11) a orientaci (12) gradientu (obr. 18). Ty rozdělíme podle úhlů do 36 skupin (binů) a ze skupin sestojíme *histogram orientací* (*orientation histogram*). Každý vzorek, přidaný do histogramu, je vážený svojí hodnotou gradientu a Gaussovým váženým kruhovým oknem se σ s hodnotou 1.5 krát hodnota klíčového bodu. Na (obr. 19) dosáhl histogram vrcholu mezi 350-359 stupni. Klíčovému bodu je tedy přiřazena orientace 36 (třicátý šestý bin). Nejvyšší hodnoty v *histogramu orientací* odpovídají dominantním směrům místních gradientů. Ze všech ostatních vrcholů, které dosáhly alespoň 80% velikosti nejvyššího vrcholu, jsou vytvořeny nové klíčové body. Ty mají stejnou pozici jako původní, a také stejnou hodnotu. Bodů, kterým je přidělena stejná pozice, je asi 15 %. Jedná se ale o důležité body, neboť podstatně zvyšují kvalitu rozpoznání. Pro přesnější určení pozice v histogramu (pozice v něm udává orientaci) sestojíme parabolu ze tří nejbližších histogramových

hodnot, nejbližších ke každému vrcholu. Z nich pak provedeme interpolaci vrcholu, abychom určili přesnější orientaci.



Obrázek 18: Hodnoty a orientace gradientů (Převzato z [13])



Obrázek 19: Histogram orientací
- 36 binů po 10 stupních (Převzato z [14])

5.4.4 Keypoint descriptor

Pro každý klíčový bod spočítáme lokální gradienty v obraze v jeho daném přiblížení. Ty jsou pak uloženy tak, aby se daly použít pro jeho opětovné rozpoznání i při špatných světelných podmínkách nebo deformaci obrazu. Deskriptory klíčových bodů jsou hodně rozdílné, proto je možné nalézt shodu i při porovnávání nad rozsáhlejší databází se slušnou pravděpodobností.

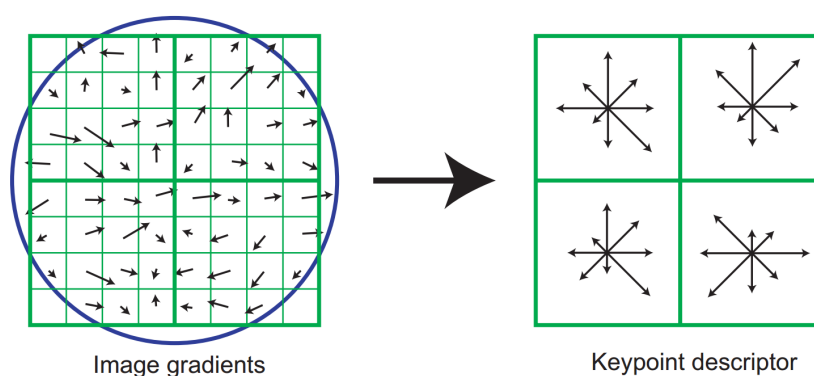
Pro lepší výkon jsou gradienty předpočítány pro všechny úrovně pyramid s klíčovými body. Gradienty jsou znázorněny na obrázku 21 v levé části malými šipkami pro každou vzorkovanou pozici v okolí klíčového bodu.

Pro určení velikosti gradientu pro každý klíčový bod použijeme Gaussovu váženou funkci se σ rovno jedné polovině šířky okna deskriptoru. Tím se plynule zmenší vliv bodů vzdálenějších od centra deskriptoru. Okolí deskriptoru rozdělíme do oblastí 4x4 (na obrázku je znázorněno pouze 2x2) a pro každou oblast vytvoříme histogram, který rozděljuje orientace gradientů do

osmi směrů a znázorníme jej pomocí šipek s velikostí podle hodnot gradientů. Vznikne nám tedy 16 histogramů, každý s osmi biny, a ty tvoří 128 prvkový „*feature vektor*“ pro každý klíčový bod.

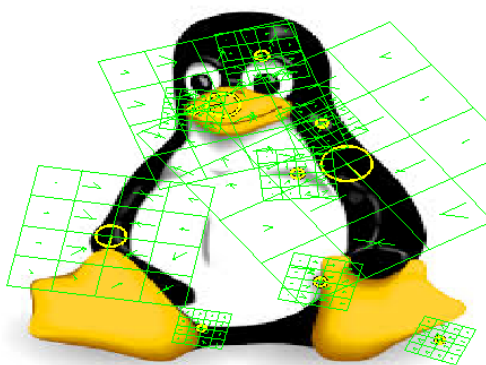
Nakonec je vektor upraven, aby se snížil vliv špatných světelných podmínek. Vektor nejprve normalizujeme, aby se zabránilo vlivu kontrastu při rozpoznávání. Jas, který přidává všem pixelům v obraze konstantu, nebude mít přílišný vliv na hodnoty gradientů, protože ty se počítají na základě rozdílů pixelů.

Deskriptoru příliš nevdá afinní změny osvětlení, nicméně má problém s nelineárními změnami. Ty vznikají například při osvětlení 3D objektů a mohou způsobit změny velikostí gradientů, ale na orientaci nemají tak velký vliv. Můžeme tedy omezit vliv velkých gradientů na velikost 0.2 (hodnota byla zjištěna experimentálně [4], a poté vektor znovu normalizovat.



Obrázek 20: Deskriptor klíčového bodu

- obrázek obsahuje deskriptor 2x2, který je spočítán z pole 8x8, zatímco v textu se popisuje deskriptor 4x4 spočtený z pole 16x16 (Převzato z [4])



Obrázek 21: SIFT klíčové body (Převzato z [15])

5.5 SLAM trackování

Má sice hodně společného s rozpoznáním „features“ v obraze, ale je natolik specifické, že si zaslouží speciální zařazení. Rozdíl oproti předchozí skupině je v tom, že se primárně nehledá souřadnice trackovaného obrazu/předmětu, ale souřadnice kamery. Navíc při využití této metody bude potřeba využít gyroskop nebo další senzory, aby poloha byla i při prudším pohybu stabilní. Pro SLAM trackování je předem nutné naskenovat okolí, kde se bude toto trackování využívat. SLAM trackování se používá také pro roboty, kteří potřebují naskenovat své okolí a zjistit svoji aktuální polohu v něm a průběžně okolí i svoji polohu aktualizovat. Tomuto způsobu se podrobněji věnovat nebudu, ale velmi pravděpodobně také využívá vyhledání feature bodů v obraze.

To může být náročné, ale využití je ohromné. Například pro určení polohy v nákupních centrech dnes zřejmě neexistuje natolik přesná technologie, která by dokázala lokalizovat mobilní telefon. Při použití iBeaconů poloha nebývá zcela přesná a údržba je nákladná. Pro hry také existuje spousta využití. S použitím HUD by se daly vytvořit nejrůznější arény „pokročilejších laser game“. Tento typ trackování řeší na dnes vysoké úrovni projekt Tango. 3.5

6 SDK pro práci s AR

SDK podporujících Android je celá řada a není snadné určit univerzální parametry, podle kterých je rozřídít. Navíc jsou stále ve vývoji, takže mezi verzemi se liší kvalita trackování i množství funkcí. Například Metaio bylo dříve (2013) v možnostech trackování zřejmě nejlepší, ale Vuforia ji od verze 4 mnohem lépe konkuruje. V kvalitě 2D trackování, které se potřebuje zřejmě nejčastěji si jsou SDK velmi podobné. Aktuálně mezi nimi zřejmě vyniká Kudan, ale lze očekávat že v novějších verzích se může vše změnit.

Kdybych měl nyní volit jednu z knihoven, tak bych se snažil preferovat ARToolkit. Je možné jej rozšířit, pokud je potřeba, je bezplatný a na mnoho AR aplikací by měl postačovat. U komerčních knihoven bychom měli obvykle počítat s cenou přinejmenším \$1000 za rok.

U renderovací části SDK by nás v budoucnu mohlo zajímat, jestli knihovny podporují vulkan technologii namísto OpenGL, ale pro aktuální aplikace to zatím nemá využití. Tím, že se jedná o dynamickou IT oblast, je ale vhodné sledovat aktuální trendy a prohledat několik fór, které se problematikou zabývají, a zjistit zda neexistuje kvalitnější řešení.

Od SDK pro AR dostupných pro Android často požadujeme:

- **Trackování**

markerů, podkladů, 3D objektů

- **Renderování**

3D modelů, podpora „video overlay“

- **Rozšiřitelnost**

IDE, licence

- **Polohu**

GPS, kompas, akcelerometr, gyroskop

6.1 Argon

Jde o webový prohlížeč podporující AR. Cílem projektu je nabídnout AR širšímu množství vývojářů. Ti mohou AR prvky zakomponovat pomocí otevřeného API do svého webu. Toto řešení by mohlo být vhodné zejména pro webové vývojáře. [8]

- markery - používá trackovací systém Vuforia
- features - používá trackovací systém Vuforia
- 360°panorama
- podporuje GPS

- lze zobrazit 2D grafiku podporující html a javascript

6.2 ARToolKit

Je možná nejrozšířenější. Má velkou komunitu, která jej neustále zdokonaluje a používá. Lze jej stáhnout z GitHubu a zapojit se. Je pod LGPL licencí, takže se není třeba obávat problémů do budoucna jako u jiných řešení (jmenovitě Vuforia). Umožňuje použití čistě Javy ale i NDK a podporuje také Unity. [26]

- podporuje asi vše mimo trackování 3D modelů a trackování obličeje, ale v budoucnu je, podle mého názoru, alespoň trackování 3D modelů očekávatelné

6.3 Mixare

Jedná se o další AR browser, jehož výhodou je GPLv3 licence. Nepodporuje žádný typ trackování a umožňuje použít pouze navigaci pomocí GPS. [28]

- Pouze GPS

6.4 Kudan

Nabízí pravděpodobně nejkvalitnější trackování. To může být díky tomu, že se jedná o novější knihovnu. Podle mého názoru docílili lepšího trackování především pomocí tzv. „extended tracking“. Jedná se o schopnost neztratit pozici objektu i v případě, že se nepodařilo nalézt žádné feature body nebo jen jejich malou část. To nemusí být snadné a zřejmě k tomu používají i senzory, protože jsou schopni „sledovat“ objekty, i když jsou zcela mimo kameru nebo pokud je obraz velmi roztřesený. Zvládne také rozpoznání objektu z poměrně veliké vzdálenosti oproti konkurenci. Stabilitu obrazu demonstrují tvůrci na mnoha Youtube videích, kde se srovnávají s velkou částí konkurence. Nevýhodou může být absence bezplatné licence, pokud se rozhodneme aplikaci releasovat. [27]

- markery
- features
- trackování 3D modelů
- SLAM trackování
- GPS
- video overlay
- virtuální tlačítka

6.5 Vuforia

Je velmi rozšířená a díky velké komunitě a podpoře vývojářů může být dobrou volbou. Její nevýhodou může být cena. Tvůrci začali s nižší cenou oproti konkurenci, ale postupně s novými verzemi licenci měnili. Pokud ji ale chceme používat jen pro základní rozpoznávání a nevadí nám průhledné logo vlevo dole, pak je knihovna zdarma i pro komerční účely a to na spoustu projektů plně dostačuje. Tato knihovna je využita pro ukázkovou aplikaci. [29]

- markery
- features
- cloudové rozpoznávání
- trackování 3D modelů
- SLAM trackování
- GPS
- video overlay
- virtuální tlačítka

6.6 Wikitude

Nabízí velké množství trackovacích metod. Je také velmi rozšířená, takže je dobré si ji uvést. Umožňuje relativně snadné propojení s PhoneGapem nebo Xamarinem. [30]

- markery
- features
- trackování 3D modelů
- trackování obličejů
- trackování QR a bar kódů
- GPS
- cloudové rozpoznávání
- video overlay
- virtuální tlačítka

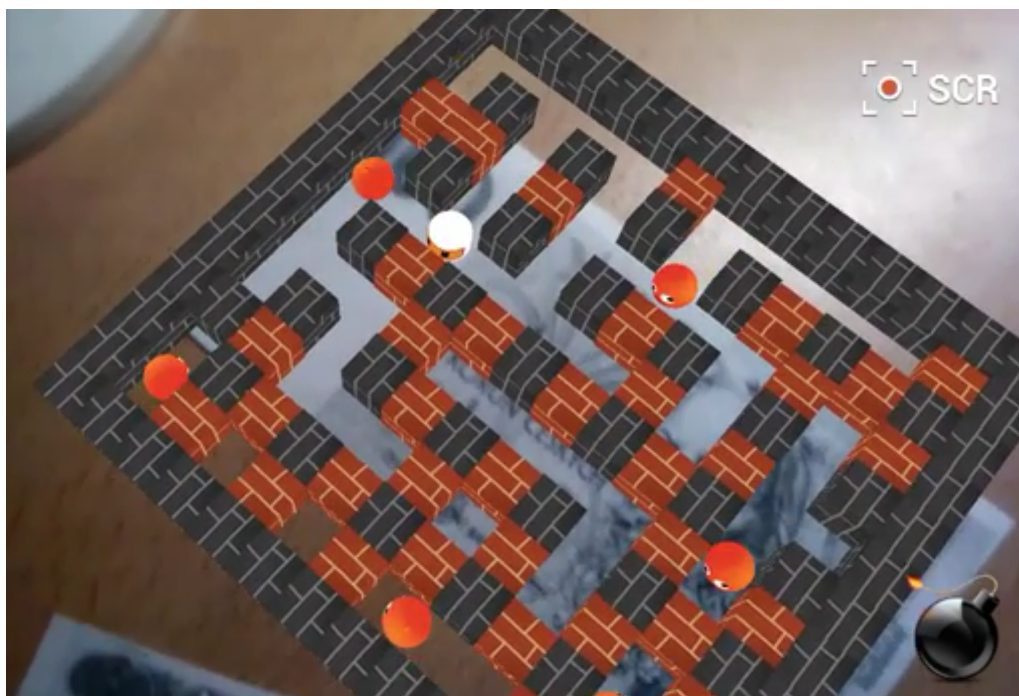
7 Návrh a Implementace hry

Tato kapitola je věnována návrhu a později implementaci hry Bomberman. Je v ní také podrobněji popsáno jak začít s knihovnou Vuforia při vývoji AR aplikace a také jak vytvořit 3D modely pro takovou aplikaci.

7.1 Inspirace

Jedná se o jednu z nejnáročnějších částí. Špatný nápad odsoudí celou hru ještě před její realizací. Odhadnout dobu pro fázi inspirace není snadné. Můžeme dostat výborný nápad ve vteřině nebo také nikdy. Poté, co se to podaří, nebo si to alespoň myslíme, je dobré zkontrolovat, jestli podobný nápad neměli již před námi. Pokud ano, pak máme výhodu a můžeme se poučit z předchozích chyb, pokud ne, pak máme zase konkurenční výhodu. Než jsem začal s vývojem hry, prostudoval jsem si mnoho AR her, abych lépe věděl, co od AR očekávat, a také jsem věnoval i chvíli Bombermanovi. Většina AR her obsahovala shodně jeden marker a na něm se odehrával smysl celé hry ve 3D. Moc se toho nezměnilo ani dodnes. Proto mě napadlo udělat něco, co se velmi blíží výsledné hře (obr. 22).

Později jsem dostal ještě další nápady, kdy by hru hráli lidé z reálného světa a bomby pokládali do ulic města podle GPS. Exploze by pak zabíjela na základě rádia bomby. Otázka pak je, do jaké míry by to bylo hratelné, ale zřejmě by to také příliš neodporovalo AR trendům. Šlo by také načíst mapu města a exploze by směřovaly pouze do daných ulic, aby se hra více podobala Bombermanovi. Toto řešení by však bylo implementačně výrazně náročnější.



Obrázek 22: Bomberman AR

7.2 Volba použitých technologií

Pro hru Bomberman jsem si stanovil pro SDK následující požadavky:

- podpora Androidu
- bezplatná licence i pro vydání na Google Play
- nadprůměrná kvalita trackování
- očekávaná podpora do budoucna
- široká komunita
- pokud možno využití již pro mě známých technologií

Při hledání vhodného SDK jsem narazil poměrně brzy na knihovnu Vuforia. Líbila se mi její rozšířenost a i dnes jí oproti konkurenčním knihovnám nic zásadního neschází. Vuforia mi od začátku připadala jako plně dostačující ve všech parametrech.

Během vývoje nastal pouze problém s nekompatibilitou s novější verzí, protože Vuforia vydala větší aktualizaci SDK 4.0 a starší verze, která měla problémy s novými telefony, již nebyla podporována. Byla možnost ponechat kód v C++, ale i tak by bylo potřeba značnou část opravit nebo nově Vuforia podporovala také Javu. Na webu Vuforie byly příklady již pouze v Javě, takže mi použití Javy přišlo jako snadnější možnost. V nové verzi knihovna požaduje autentikaci, ale naštěstí již nebrání spuštění aplikace, pokud je uživatel offline.

Pro vývoj jsem zvolil Eclipse IDE, aktuálně projekt běží na verzi 4.5.

Vuforia SDK je ve verzi 4.0.103.

7.3 Změny oproti původní Dyna Blaster verzi

7.3.1 Grafické zpracování

Starší verze obsahovala velké pixely, jako všechny hry té doby a bylo potřeba rozhodnout, jestli vyrobit 2D remake hry nebo použít hladké nebo kostičkované 3D modely a podle toho pak také textury.

7.3.2 Dopady na hratelnost

Od počátku jsem se snažil o klon hry, takže byl jediným rozdílem pouze způsob ovládání přes dotykový displej. Později jsem experimentoval s alternativním ovládáním založeném na sledování více markerů. Zkoušel jsem do hry druhým markerem přidávat další protivníky, to ale hratelnostně nebylo příliš použitelné. Vznikl z toho nápad pro alternativní ovládání hry joystickem umístěným mimo displej, který by byl oním markerem a fungoval by stejně jako joystick na displeji. Bomba by se dala vyřešit třetím markerem nebo Vuforia podporuje také rozpoznání

kliknutí prstem - virtual buttons. Takové ovládání se určitě v budoucnu může používat u AR brýlí, které nemohou používat dotykový displej.

Dále jsem se pokoušel využít akcelerometru, který by mohl ovlivnit gravitaci působící na bomby a protivníky. To nakonec uplatnění v Bombermanovi nenašlo. Bylo to ale zajímavé, takže jsem tento způsob využil pro ukázkou hry s balancováním kuličky v labyrintu (viz. kapitola 7.6). Dalším pokusem bylo využití 3D zobrazení. To, že by monstra ve hře dostala schopnost létat, by mohlo hru odlišit od svého předchůdce. Létající monstra by nutila hráče k naklánění displeje, protože při pohledu ze shora není z-tová pozice dobře patrná. Ze stejného důvodu je ve hře také menší monstrum, které je díky své velikosti a tomu, že se pohybuje podél stěn špatně vidět. To také nutí hráče dívat se z více úhlů.

7.4 Knihovna Vuforia

V této části se pokusím ukázat, jak se s touto knihovnou pracuje. V rámci této práce používám knihovnu ve verzi 4.0.103, ale nyní je dostupná již verze 5.5.9 a také bude vhodnější použít spíše Android Studio, které už je od roku 2015 oficiálně doporučováno pro vývoj Android aplikací. Instalace Android Studia i spuštění základní Vuforia aplikace je poměrně jednoduché, protože cesty a SDK se nastaví víceméně automaticky.

Výhodou oproti mé starší verzi je možnost zapnutí aplikace bez internetového připojení. Vuforia totiž při svém prvním spuštění aplikace kontrolovala použití licenčního klíče, a přestože jinak aplikace mohla fungovat offline, nedovolila její spuštění. Od verze 4 tuto kontrolu Vuforia zavedla a sklidila veliký odpor vývojářů. Proto ji v dalších verzích odebrali.

7.4.1 Založení Vuforia projektu

Nejprve si budeme muset stáhnout a nainstalovat samotné Android Studio ². Dále budeme potřebovat založit si developerský účet ³, který je nezbytný k získání licenčního klíče.

V dalším kroku v sekci downloads stáhneme Vuforia SDK a příklady pro Android⁴, které otevřeme v Android Studiu. Pak jen zbývá nastavit cestu k VUFORIA_SDK_DIR v otevřeném projektu s příklady. Tuto proměnnou najdeme v `app/src/build.gradle`.

Nakonec ve třídě `com.vuforia.samples.SampleApplication` přidáme metodě `setInitParameters(...)` jako třetí parametr 380 znaků dlouhý klíč vygenerovaný na developerském portálu Vuforia a můžeme spustit AR aplikaci s pomocí Vuforia knihovny.

7.4.2 Práce s knihovnou

Každý z příkladů obsahuje dvě třídy. Pro vykreslení vlastního 3D modelu je zajímavá metoda `initApplicationAR()` (Výpis 1), která nastaví `mGLView.setRenderer(mRenderer); mRenderer`

²<https://developer.android.com/studio>

³<https://developer.vuforia.com>

⁴Obojí ve verzi 5.5.9. Novější se mohou nepatrně lišit.

je typu `GLSurfaceView.Renderer` a implementuje metodu `onDrawFrame(GL10 gl)`, ve které pak vykreslíme pomocí OpenGL naši připravenou scénu a nalezneme *trackables* pro umístění scény do nalezených souřadnic. Matice nalezených *trackables* získáme stejně jako v této ukázce - Výpis 2.

```
GLSurfaceView mGlowView;  
  
// Initializes AR application components.  
private void initApplicationAR()  
{  
    // Create OpenGL ES view:  
    int depthSize = 16;  
    int stencilSize = 0;  
    boolean translucent = Vuforia.requiresAlpha();  
  
    mGlowView = new SampleApplicationGLView(this);  
    mGlowView.init(translucent, depthSize, stencilSize);  
  
    mRenderer = new ImageTargetRenderer(this, vuforiaAppSession);  
    mRenderer.setTextures(mTextures);  
    mGlowView.setRenderer(mRenderer);  
}
```

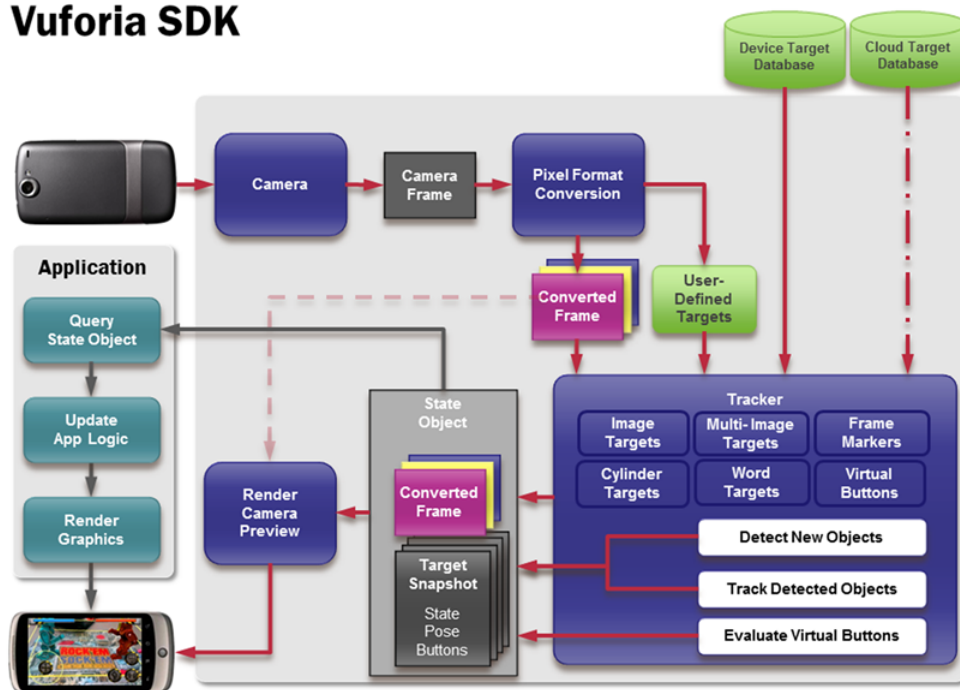
Výpis 1: Část kódu pro lepší orientaci v ukázkových aplikacích Vuforia.

Vuforia podporuje také cloudové trackování, kdy umožňuje nahrát i více než 100 000 trackables do svého cloudu. Tím se od ostatních snaží odlišit a za cloudové rozpoznání si nechávají platit. Celý proces načítání a rozpoznávání trackables knihovnou Vuforia je přehledně znázorněn na obrázku 23. Kamera telefonu nejprve zachytí snímek a ten je možné použít jako vlastní *target* nebo jej předat *trackeru* k rozpoznání. Tracker může rozpoznávat pomocí tří targetů:

1. **user defined** - vytvořených uživatelem po instalaci aplikace
2. **devide database** - uložených v databázi v zařízení: nahraných přes vuforia portál a přidáných k aplikaci (obr. 26)
3. **cloud database** - uložených v cloudu: je potřeba internetové spojení k rozpoznání

Snímek z fotoaparátu se předá přímo k zobrazení na telefon a přes něj aplikace překreslí virtuální obsah. V obrázku můžeme vidět také *virtual buttons* - jde o trackables, které reagují na „kliknutí“. Zkoušel jsem je ve verzi Vuforia 2.6 a nebyly příliš přesné. Šlo spíše o detekci překrytí než o reakci na stisknutí tlačítka.

Vuforia SDK



Obrázek 23: Data flow diagram - Vuforia SDK

7.5 Vytvoření herních modelů

V této části se podíváme podrobněji na jeden ze způsobů, jak je možné vytvořit 3D modely pro AR aplikaci pomocí volně dostupných nástrojů.

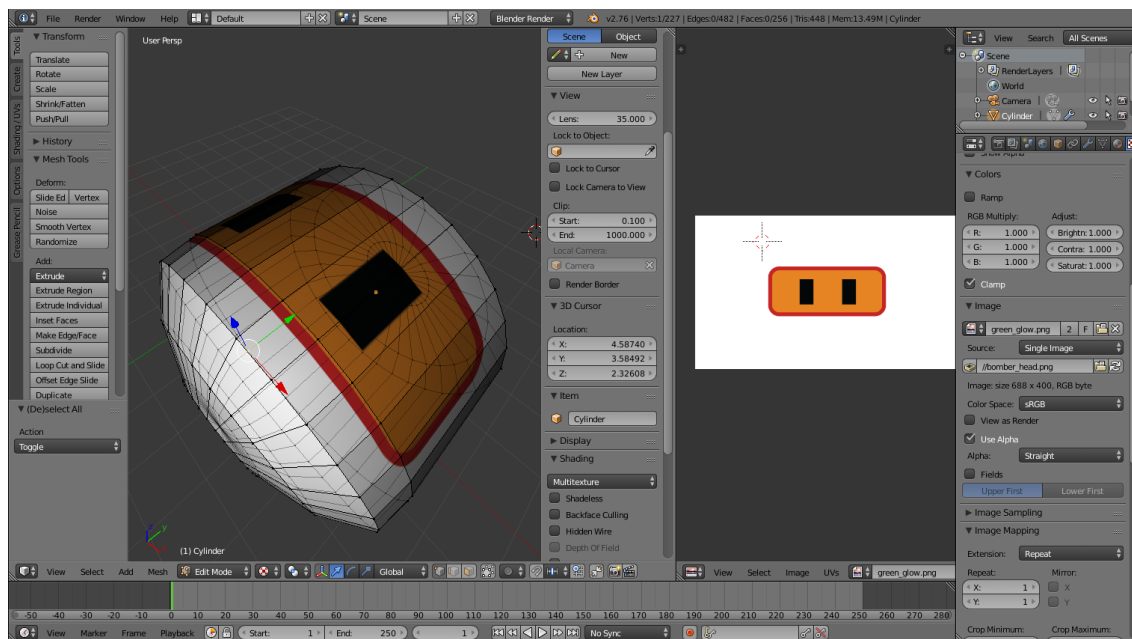
7.5.1 Volba 3D editoru

Na výběr dnes máme z mnoha nástrojů. Pro 3D modely jsem zvolil osvědčený Blender. Výhodou Blenderu ⁵ je bezplatná licence i pro komerční účely a k tomu má širokou komunitu. Umožňuje také export do mnoha formátů, takže by se nemělo stát, že změnou AR technologie by pracně vytvořené modely byly nepoužitelné. Nevýhodou je trochu nepřehledné prostředí, které se s každou verzí mění a rozhodně nejde označit jako intuitivní. V mnoha ohledech může být proto vhodnější volbou SketchUp, zvláště pro začátečníky, kteří s Blenderem nemají zkušenosti.

7.5.2 Vytvoření textur

Tady volba také není snadná, pokud nechceme použít placený nástroj. Základní otázka asi je, zda dáváme přednost bitmapové nebo vektorové grafice. Pro bitmapovou zřejmě neuděláme chybu s GIMPem. Mi vyhovuje více vektorová grafika, proto jsem zvolil osvědčený Inkscape.

⁵www.blender.com



Obrázek 24: Blender

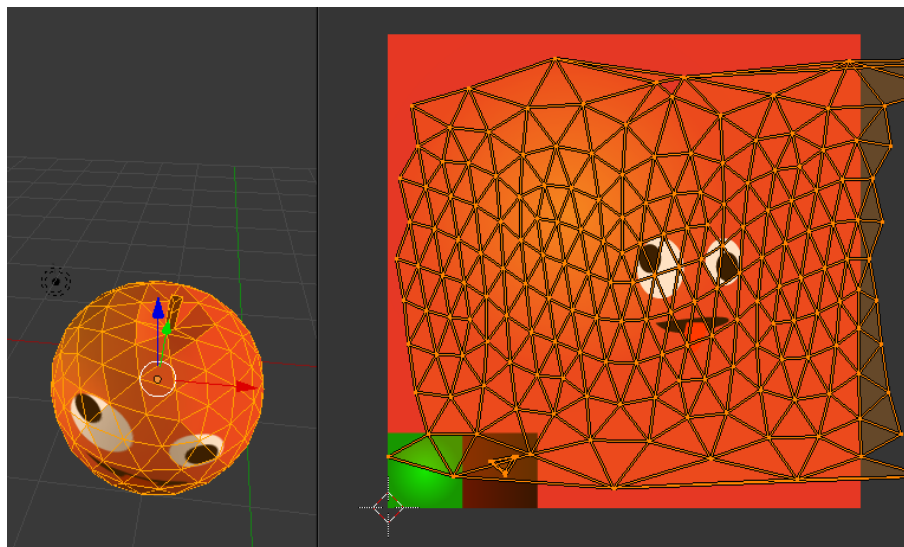
7.5.3 Načtení v OpenGL

Pro načtení modelu v OpenGL v C++ jsem použil řešení, které používala Vuforia komunita od Heiko Behrense [2]. Jedná se o skript v Perlu, který převádí soubory *.obj* do hlavičkových souborů v C/C++ a usnadňuje jejich snadné načtení. Později nebyl problém také s exportováním do Javy s několika drobnými úpravami. Skript má problém s exportem komplexnějších objektů nebo s těmi, které obsahují více textur. Pro více textur je potřeba v Blenderu použít ruční mapování, které Blender dobře podporuje obr. 25. Pro načtení v Javě jej brzdí omezení pro velikost vygenerovaného bytekódu 64k z jedné metody. Skript totiž zaokrouhluje hodnoty na příliš mnoho desetinných míst. Pro velké objekty je proto potřeba načtení dat ze souboru nebo jejich rozdělení.

```
for (int tIdx = 0; tIdx < state.getNumTrackableResults(); tIdx++) {

    TrackableResult result = state.getTrackableResult(tIdx);
    Trackable trackable = result.getTrackable();
    printUserData(trackable);
    Matrix44F modelViewMatrix_Vuforia = Tool
        .convertPose2GLMatrix(result.getPose());
    modelViewMatrix = modelViewMatrix_Vuforia.getData();

    render(modelViewMatrix);
}
```

Obrázek 25: Využití jedné textury pro více barev

Výpis 2: Získání projekční matice pro rozpoznané „*trackables*“ (plochy) s pomocí knihovny Vuforia

7.5.4 Vytvoření „*trackables*“

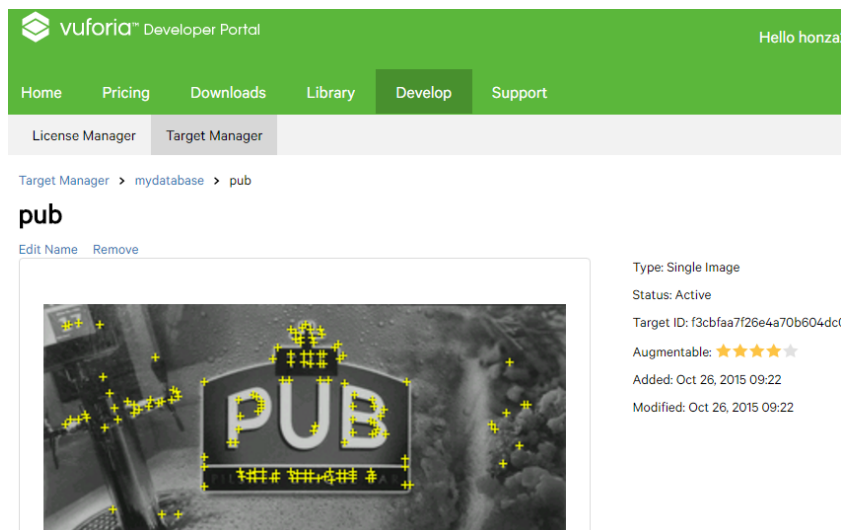
Tato část je do jisté míry „černá skříňka“, protože Vuforia nesdílí, jakým způsobem přesně vyhledává „feature points“ v obraze. Je možné, že využívají výše popsany SIFT nebo v mnoha případech rychlejší SURF. Je také možné, že používají nějaký nový algoritmus, nebo že algoritmy postupně vylepšují a s verzemi obměňují. Vuforia podporuje také 3D trackování, takže mohou používat i kombinaci rozpoznávacích algoritmů. Ve webovém rozhraní pro vývojáře si je potřeba nahrát vhodný obrázek pro trackování a Vuforia jej ohodnotí jednou až pěti hvězdičkami (obr. 26). Obrázek naznačuje, že knihovna trochu podle očekávání pracuje pouze s černobílými obrázky.

Databázi takto vložených obrázků je poté možno stáhnout jako 2 soubory `.dat` `.xml` a připojit k aplikaci. Aplikace může používat i více takovýchto databází.

7.5.5 Návrh herní části

Pokud chceme zachovat hrátelnost ze staršího Bombermana, je třeba navrhnout hru obdobně jako je v původní hře. Před zahájením hry se vytvoří instance hry `Game` a hra inicializuje metodou `initLevel()`. Vytvoří se herní deska šířky `boardWidth` a s výškou `boardHeight`.

Do herní desky jsou vloženy zdi `Wall`, které mohou být dvojího typu `WallType`. Cihlové (`BRICK`) lze rozbít bombou a naopak kamenné (`STONE`) bombou zničit nelze. Zdi mají souřadnice v celých číslech a jsou na hrací desce umístěny do mřížky. Kamenné zdi se umísťují vždy ob jeden řádek a



Obrázek 26: Vývojářská část portálu Vuforia - nahrání „trackables“

sloupec a také do patra nastavením proměnné `posZ` z 0 na 1. Cihlové zdi se umístí mezi kamenné v prvním patře a ve druhém patře pouze pokud se pod nimi již cihlová zeď nachází. Pokud by cihlová zeď byla pouze ve druhém patře, mohl by vzniknout problém s jejím rozbitím bombou. Zdi pak mohou vypadat podobně jako na obrázku 27.

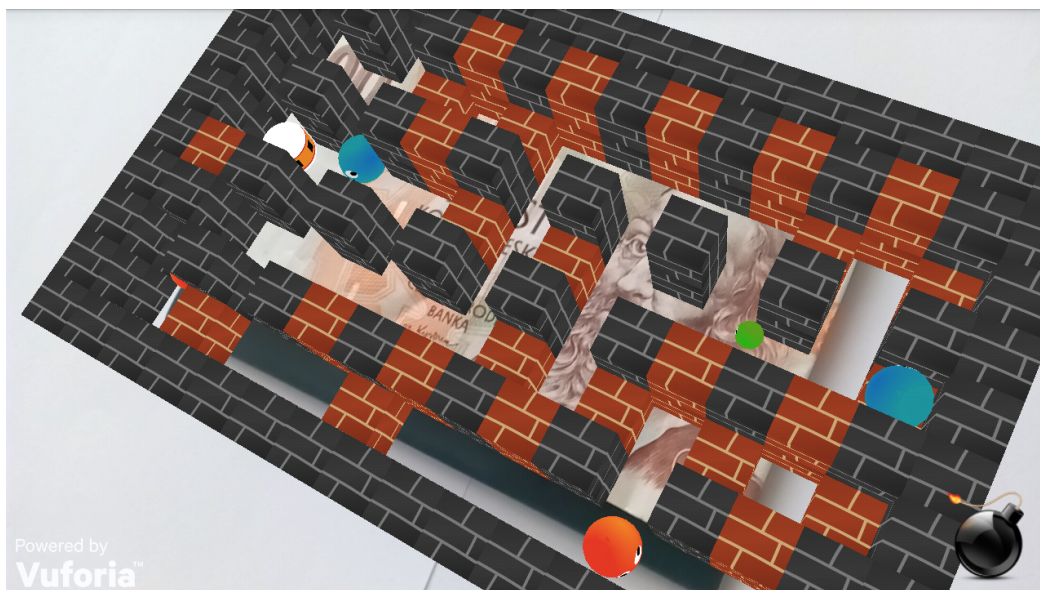
Dále jsou do scény umístěny 3 typy monster:

- **Oranžové - MadApple** : to se chová očekávaně, mění směr pouze po nárazu do zdi
- **Zelené - MadApple** : liší se hodnotou `color`. Jinak jde o menší verzi oranžového výše. Je hůře vidět, protože se drží blízko zdi
- **Modré - FlyingMonster** : liší se schopností měnit výšku. Změnit ji může jen, pokud je zcela nahoře nebo dole. Pokud je nahoře, tak je pro hráče neškodné.

Návrh bere ohled na rozšíření o multiplayer, kde je potřeba si pamatovat, která bomba je od kterého hráče. Takže object `Bomb` má vazbu na hráče, který bombu položil a z objektu `Player` bomba získá informaci o svém dosahu (`bomRadius`). To se hodí i při hře jednoho hráče, protože pokud hráč vezme bonus pro zvýšení dosahu bomby, pak se `radius` bomby zvýší i u bomb již položených. Po explozi bomby se vytvoří v horizontálním a vertikálním směru objekty (`Explosion`) podle `radius` bomby, pokud nebyla v cestě zeď. Každá (`Explosion`) má nastavený velmi krátký čas, po jehož uplynutí zmizí.

7.5.6 Návrh uživatelského rozhraní

Ovládání hry jako je Bomberman je na malém displeji poměrně problematické. Jde o velmi rychlou hru, kde je potřeba ovládat pohyb Bombermana a pokládání bomb. Ideální je proto použití obou rukou nebo akcelerometru, které ale u AR aplikace nebylo možné. Ovládání ve stylu



Obrázek 27: Ukázka ze hry Bomberman, promítnutá na českou bankovku.

šipek jsem brzy vyřadil, protože hráč nevidí pořádně pod své prsty a při hře se málokdy treťí v rychlosti na správnou šipku poslepu. Hratelnost s takovým ovládáním je pak velice špatná. Poté jsem zkoušel pevně umístěný joystick, ale ten opět nefungoval ideálně a znovu nastal problém, když hráč v rychlosti potřeboval odejít konkrétním směrem.

Jako nejlepší se mi osvědčil joystick bez pevného umístění, který funguje přibližně v jedné třetině levé části obrazovky. Střed joysticku se umístí na místo, kde se hráč dotkne, a dále funguje pomocí gest a po zvednutí prstu zmizí. Tlačítko s bombou pak vcelku logicky skončilo vpravo dole, pro druhou ruku.

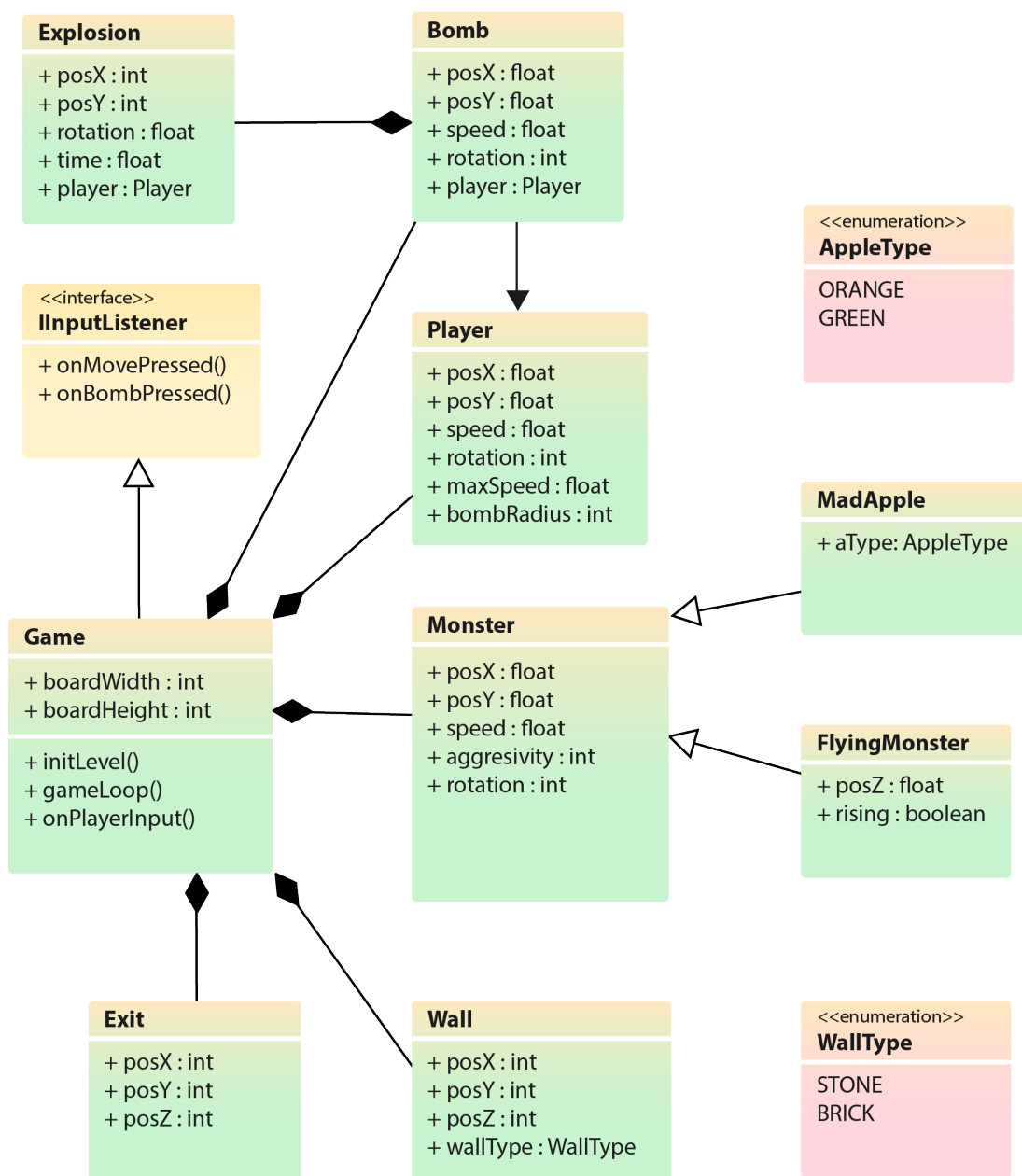
7.5.7 Vylepšení ovládání

V prvních verzích hry nastal problém s pohybem „Bombermana“. Nebylo se snadné trefit mezi zdi a podařilo se projít až na několikátý pokus. Hráč buďto mezeru přešel nebo se o ní zaseknul příliš brzy. Tato situace je zachycená na obrázku 30. Zmenšení modelu by v tomto případě nevypadalo dobře.

V pozici kdy se hráč pohybuje ve směru modré šipky, ale brání mu v tomto pohybu zeď, je potřeba vyhodnotit jeho pozici od středu stěny, o kterou se zaseknul a posouvat jej k nejbližšímu okraji stěny jak znázorňuje růžová šipka. Tato drobná změna velice zvýší hratelnost obzvláště na dotykovém displeji a navíc se hra více podobá původní verzi.

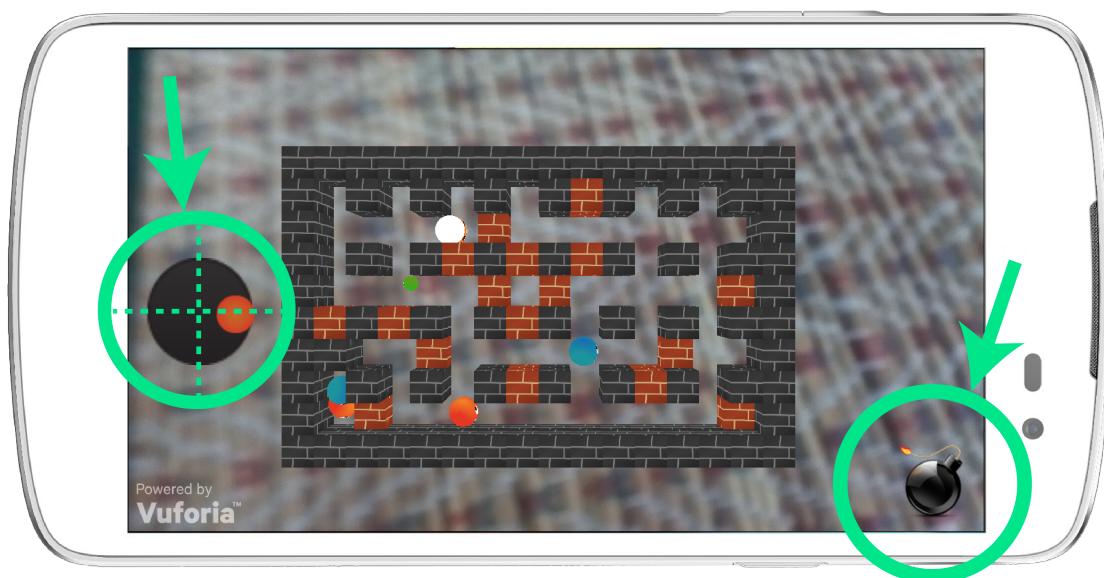
7.5.8 Otestování aplikace z pohledu AR

Testování výkonu aplikace dopadlo do jisté míry zajímavě. Veliký vliv na vykreslování snímků mělo renderování scény v OpenGL podle její náročnosti (viz. tabulka 1). Poměrně překvapivý

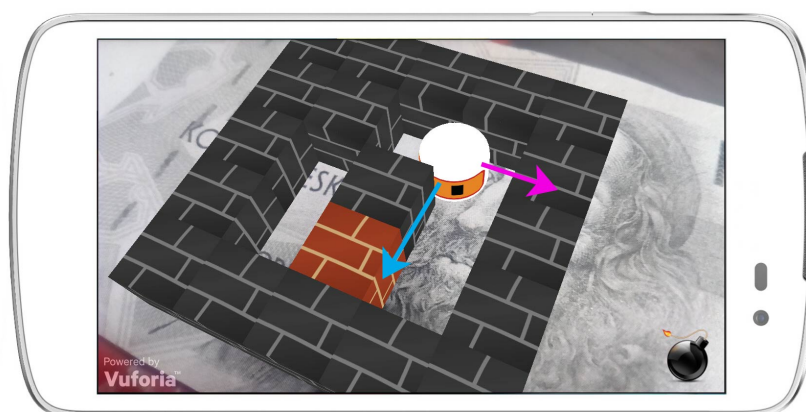


Obrázek 28: Třídní diagram pro hru Bomberman

byl vliv světla. V běžném pokojovém osvětlení, při kterém jsem prováděl záznamy do tabulky, se vykreslování pohybovalo velmi stabilně kolem 16,7 FPS. Při velmi silném, například venkovním světle nebo pod lampou nebyl problém přesáhnout 30 FPS nebo se stabilně pohybovat nad hodnotou 29 FPS. Naopak v naprosté tmě se FPS pohybuje kolem hodnot 15,3 FPS, chvílemi dokonce pod 15 FPS. Světlo má tedy ohromný vliv na výkon knihovny Vuforia. Na rovnoměrně osvětleném bílém papíře knihovna také dosahovala téměř 30 FPS, takže nalezené „feature points“



Obrázek 29: Zachycení joysticku a tlačítka pro pokládání bomb a jejich zvýraznění.



Obrázek 30: Obcházení zdí.

zřejmě nemají na výkon podstatný vliv.

Při použití knihovny Vuforia platí také podmínky z kapitoly 4.

Tabulka 1: Rychlost FPS při různém nastavení aplikace.

Rozměry herní desky	Počet monster	FPS
5x5	1	16.7
5x5	100	16.7
5x5	500	8.4
16x10	6	16.7
16x10	100	16.7
30x30	6	8.6
50x50	1	3.55
50x50	6	3.5

7.6 Hra Labyrint

Jde o další jednoduchou hru s AR prvky, která je součástí programové implementace této práce. Stejně jako Bomberman také využívá knihovnu Vuforia. Nápad k ní vznikl, když jsem se pokoušel využít akcelerometr a gyroskop ve hře Bomberman, ale hratelnosti to příliš nepřidalo. Nápad to byl ale dost zajímavý, proto jsem přemýšlel, jak jej zpracovat. Princip hry je jednoduchý, jde o dopravení kuličky do středu labyrintu. Podobných her je poměrně hodně. Tato ale využívá také polohu zařízení a snaží se eliminovat natočení telefonu a dát hráči pocit, jako by balancoval kuličkou pouze se zemskou gravitací.

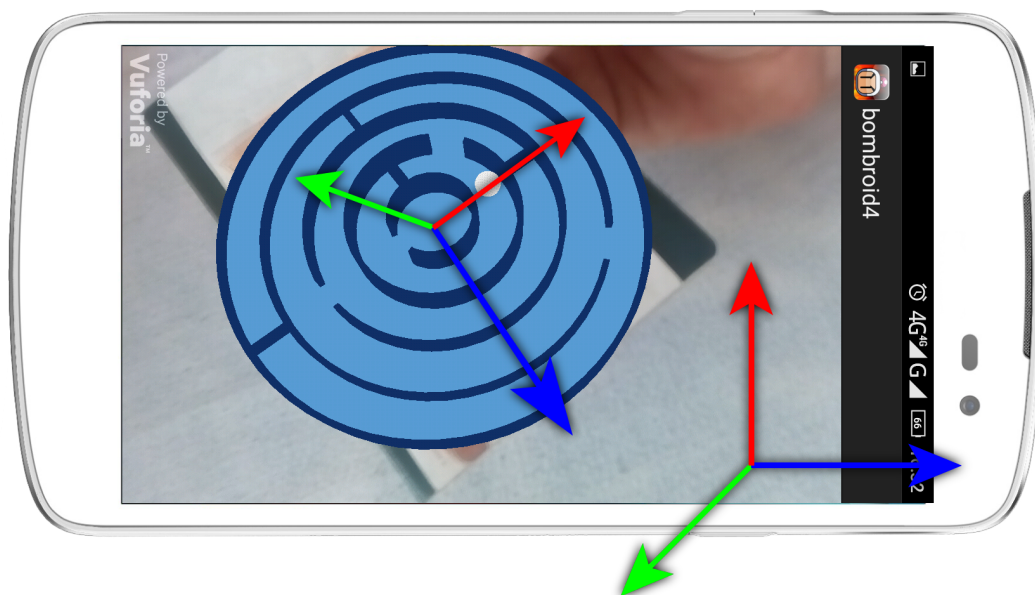
Potřebujeme tedy zjistit natočení scény vůči telefonu. Z projekční matice (Výpis kódu 2) získáme úhly natočení scény (Výpis kódu 4) a pomocí akcelerometru zjistíme aktuální natočení telefonu (Výpis kódu 3). Poté stačí úhly pouze odečíst a dostaneme aktuální naklonění podložky, ze kterého získáme zrychlení kuličky. Tím se vytvoří pohyb kuličky nezávislý na naklonění telefonu, takže hráči připadá, jako by na kuličku působila skutečná gravitace.

```
final float[] mValuesOrientation = new float[3];
final float[] mRotationMatrix = new float[9];

@Override
public void onSensorChanged(SensorEvent event) {

    if (event.sensor.getType() == Sensor.TYPE_ACCELEROMETER) {

        //obtains angles
        SensorManager.getRotationMatrix(mRotationMatrix, null, mValuesAccel,
            mValuesMagnet);
        SensorManager.getOrientation(mRotationMatrix, mValuesOrientation);
```



Obrázek 31: Labyrint - souřadnice scény a telefonu

```

        angX = (float) Util.rad2deg(mValuesOrientation[1]);
        angY = (float) Util.rad2deg(mValuesOrientation[2]);
    }
}

```

Výpis 3: Část kódu zjišťující natočení telefonu

```

//obtains angles in degrees from projection matrix
double xAngle = rad2deg(Math.asin(modelViewMatrix[8]));
double yAngle = rad2deg(Math.acos(modelViewMatrix[5]));

```

Výpis 4: Část kódu zjišťující natočení projekční matice vůči telefonu.

8 Závěr

Rozšířená realita se začne brzy rozšiřovat a uplatňovat mezi dnešními hrami a postupně nahradí herní konzole a desktopové hry. Mobilní zařízení jsou jen odrazovým můstkem pro aplikace s rozšířenou realitou a dá se brzy očekávat jejich nahrazení k tomuto účelu lépe uzpůsobenými zařízeními.

V práci jsem prozkoumal a popsal, jaké možnosti rozšířená realita nabízí a jaký má potenciál do budoucna, i když to je velmi těžké odhadnout u takto tvárné technologie. Popsal jsem základní typy algoritmů, které je možné pro rozšířenou realitu použít a uvedl k nim příklad. Podařilo se mi vyrobit hru, u které je na první pohled patrná inspirace hrou Bomberman. K tomu aby hra dosáhla vyššího počtu stažení například na Google Play, by bylo třeba vylepšit hratelnost nebo přidat počet úrovní a celkově hru více dotáhnout. V tomto stádiu se jedná spíše o ukázkou možností AR, než o zábavu na desítky hodin. Na Google Play jsem aplikaci zatím nepublikoval, k tomu bude možná vhodnější druhá hra Labyrint, která je o něco jednodušší a vhodnější pro mobilní ovládání. Dále jsem v práci prozkoumal knihovny pro rozšířenou realitu a podrobněji vyzkoušel knihovnu Vuforia, kterou jsem použil pro implementaci aplikace. Vytvořil jsem 3D modely pro hru, ale animace jsem udělal jen velmi základní. To je graficky a časově poměrně náročné. Aplikaci jsem nakonec otestoval v různých světelných podmínkách a také z hlediska výkonu.

Jan Hruška

Literatura

- [1] Wolfson , Haim J., *Geometric Hashing: An Overview*, Tel Aviv University, 1997.
- [2] Corne, Leslie, *Marker Detection for AR Applications*, <https://infi.nl/nieuws/marker-detection-for-augmented-reality-applications/>, 2010.
- [3] Behrens, Heiko, *Skript pro převod 3D modelů*, <http://heikobehrens.net/2009/08/27/obj2OpenGL/>, 2009.
- [4] Lowe, David G., *Distinctive Image Features from Scale-Invariant Keypoints*, University of British Columbia 2004.
- [5] Harris, C. and Stephens, M, *A combined corner and edge detector*, Plessey Research Roke Manor, United Kingdom 1988.
- [6] Benjamin Long, Sue Ann Seah, Tom Carter, Sriram Subramanian *Rendering volumetric haptic shapes in mid-air using ultrasound*, University of Bristol, UK 2014.
- [7] Paul Milgram *A taxonomy of Mixed Reality Visual Displays*, IEICE Transactions on Information Systems, Vol E77-D, No.12 December 1994.
- [8] Augmented Environments Lab., *AEL*, <http://ael.gatech.edu/lab/research/arweb/argon-an-open-standards-ar-web-browser/>, 2009.
- [9] *Augmented/Virtual Reality Report 2015*, Digi-Capital, <http://www.digi-capital.com/news/2015/04/augmentedvirtual-reality-to-hit-150-billion-disrupting-mobile-by-2020>, 2015.
- [10] *Zuno mobile banking 2012*, ZUNO Bank, 2012. <https://www.zuno.cz/>, 2012.
- [11] *sixthsense*, Pranav Mistry. <http://www.pranavmistry.com/projects/sixthsense>.
- [12] *SIFT: Scale invariant feature transform* by David Lowe, Silvio Savarese. http://web.eecs.umich.edu/~silvio/teaching/EECS598/lectures/lecture10_1.pdf.
- [13] *Ai Shack - SIFT*, Utkarsh Sinha. <http://aishack.in/tutorials/sift-scale-invariant-feature-transform-keypoint-orientation/>.
- [14] YU MENG and Dr. Bernard Tiddeman, *Implementing the Scale Invariant Feature Transform(SIFT) Method* University of St. Andrews.
- [15] *Image registration using robust least squares*, Ivan Bogun. <http://my.fit.edu/~ibogun2010/Tutorials/featureMatching/>, 2014.
- [16] *Microsoft www stránky*, Microsoft. <https://www.microsoft.com/microsoft-HoloLens/en-us/apps/holostudio>, 2016.

- [17] *We found 7 critical HoloLens details that Microsoft hid inside its developer docs*, Mark Hachman. <http://www.pcworld.com/article/3039822/consumer-electronics/we-found-7-critical-HoloLens-details-that-microsoft-hid-inside-its-developer-docs.html>, 2016.
- [18] *Leap Motion www stránky*, Leap Motion. <https://www.leapmotion.com/product/vr>, 2016.
- [19] *Leap Motion - Youtube*, Leap Motion. <https://www.youtube.com/watch?v=1x-eAvASIFc>, 2016.
- [20] *Projekt Minuum www stránky*, Whirlscape Inc. <http://minuum.com/>, 2015.
- [21] *Woorld game - Youtube*, Engadget. https://www.youtube.com/watch?v=vg_VSoScB_s, 2016.
- [22] *Ukázka hry Longbow od Valve - Youtube*, Nathie. https://www.youtube.com/watch?v=gwtKAXr_xWo, 2016.
- [23] *11 BEST Google Cardboard VR games / apps 2015 - Youtube*, Google Cardboard VR Apps. <https://www.youtube.com/watch?v=j5jlEZ0t9XQ>, 2015.
- [24] *Hra Zombies Everywhere!*, UselessCreations. <http://uselessiphonestuff.com/?app=zombies-everywhere-augmented-reality-apocalypse>, 2013.
- [25] *Hyper-Reality - video*, Keiichi Matsuda. <https://vimeo.com/166807261>, 2016.
- [26] *ARToolKit www stránky*, Hirokazu Kato of Nara Institute of Science and Technology. <https://artoolkit.org/>, 1999.
- [27] *Kudan www stránky*. <https://www.kudan.eu/>, 2011.
- [28] *Open Source Augmented Reality Engine*, Daniele Gobbetti, Miriam Fostini, Hannes Boran. <http://www.mixare.org/>, 2011.
- [29] *Vuforia www stránky*. <http://www.vuforia.com/>, 2010.
- [30] *Wikitude www stránky*. <http://www.wikitude.com/>, 2008.